

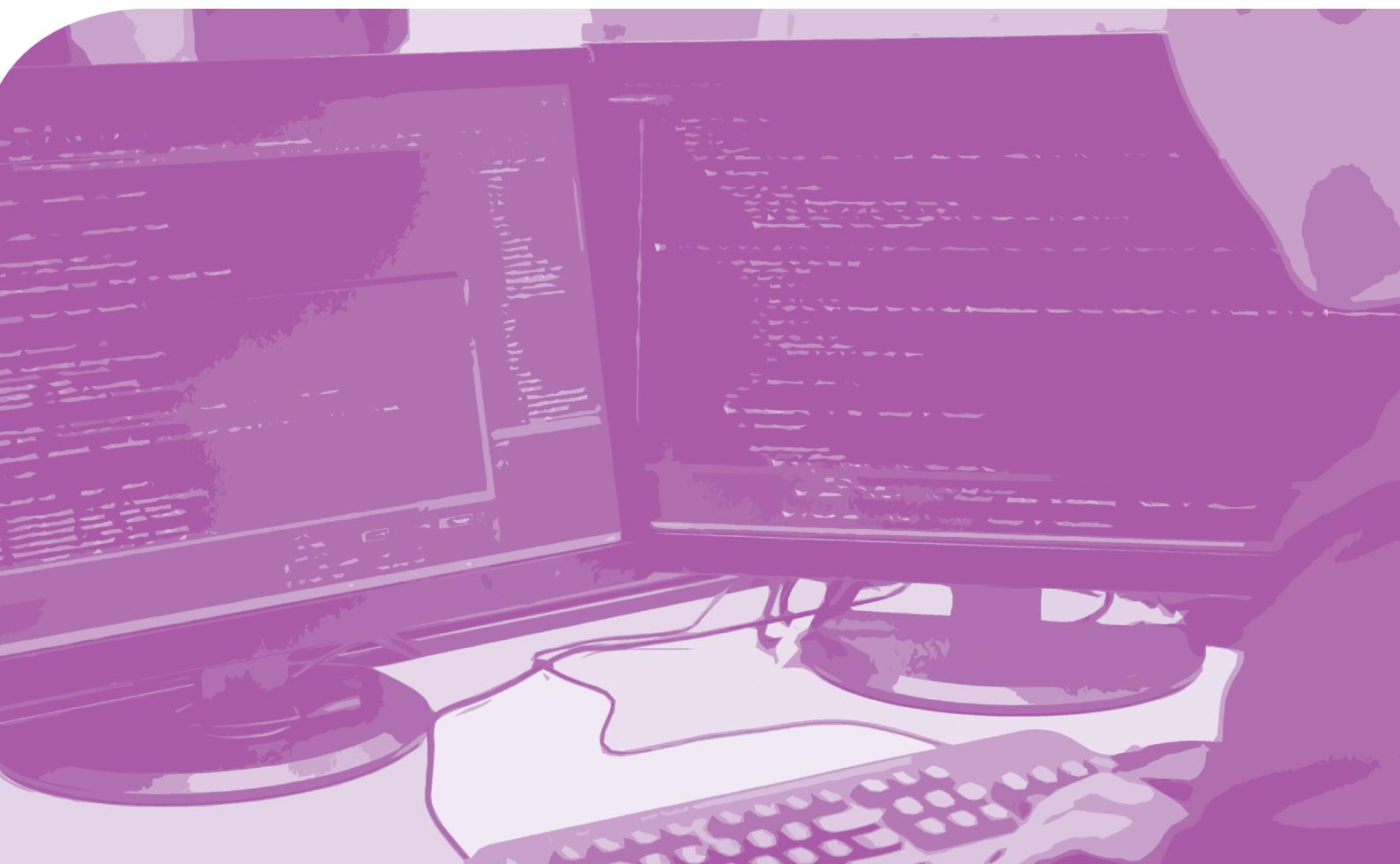
Pearson BTEC Узбекистан Уровень 4 Квалификации в
сфере

разработки программного обеспечения

Модуль 1: Введение в программирование

Учебно-методический материал

Выпуск 1



Edexcel, BTEC и LCCI квалификации

Квалификации Edexcel, BTEC и LCCI предоставлены компанией Pearson, которая является крупнейшей организацией, присуждающей сертификаты в Великобритании. Компания предлагает академические и профессиональные квалификации, признанные во всем мире и прошедшие сравнительный анализ. Для получения дополнительной информации, пожалуйста, посетите наш сайт квалификаций по адресу qualification.pearson.com. Кроме того, вы можете связаться с нами, используя контактную информацию на нашей странице по адресу qualification.pearson.com/contactus.

О Pearson

Pearson – ведущая в мире образовательная компания, которая насчитывает 35 000 сотрудников в более чем 70 странах, усердно работающих над оказанием помощи людям всех возрастов с целью достижения ощутимого прогресса в своей жизни посредством обучения. Учащийся является центром интересов компании и направленных на него действий, поскольку усовершенствование системы образования влияет на людей в целом. Вы можете узнать больше о том, как мы можем помочь вам и вашим учащимся на сайте qualification.pearson.com.

Ссылки на сторонние материалы, приведенные в этом документе, сделаны добросовестно. Pearson не одобряет, не подтверждает и не возлагает на себя ответственность за содержание материалов, которые могут быть изменены, или за любые высказанные мнения. (Материал может включать учебники, журналы и другие публикации и веб-сайты.)

Вся информация в этом документе верна на момент публикации.

ISBN 978 1 446 96279 4

Все материалы в этой публикации являются собственностью
© Pearson Education Limited 2019

Содержание

Введение	1
Модуль 1: Введение в программирование	3
Руководство по обучению	3
Схемы работы	9
План урока	39

Введение

Данный ресурсный буклете является дополнением к квалификациям ВТЕС Узбекистан Уровень 4 в сфере разработки программного обеспечения. В ресурсном буклете изложены направление обучения и предмет оценки. В данном руководстве вы найдете предложения и идеи относительно эффективного обучения и оценивания.

Также в буклете приведены идеи для преподавания и обучения, в том числе практические занятия, практические сценарии, способы вовлечения работодателей в процесс изучения и управление самостоятельным обучением, а также способы оценки. Буклете также показывает, как содержание спецификации может работать на практике, и он вдохновит вас задуматься о различных способах повышения квалификации.

Данный буклете предоставляет вам:

- руководство о преподавании модулей в рамках квалификации;
- рекомендуемые ресурсы для поддержки изучения модулей в рамках квалификации;
- схемы работ, которые показывают темы, виды деятельности и оценки, охватываемые во всех модулях в рамках квалификации;
- планы уроков с подробным руководством о том, как проводить уроки по модулям.

Информация в данном буклете собрана преподавателями, которые непосредственно связаны с разработкой квалификаций и поэтому понимают проблемы поиска новых и интересных способов преподавания в рамках квалификаций ВТЕС.

В руководстве обучения, приведенном в этом буклете, содержится информация о том, что необходимо учитывать при планировании изучения в рамках квалификации. В руководстве приведена информация о:

- структуре вашей квалификации;
- о том, как вы можете построить квалификацию для ваших учеников;
- предложения о том, как вы можете связаться с соответствующими работодателями;
- информация о других вспомогательных материалах и доступных ресурсах.

Мы предоставили вам руководство по обучению для каждого модуля. Оно включает в себя предложения по ознакомлению с учебными целями и содержанием модуля, а также идеи для интересных и разнообразных занятий. Вы также найдете советы и идеи о том, как планировать и выполнять свои самостоятельные работы.

Мы включили перечень тщательно отобранных ресурсов для каждого блока. Данные ресурсы содержат предложения касательно литературы, веб-сайтов и видео, которые вы можете порекомендовать своим учащимся и / или которые вы можете использовать в качестве дополнения к изучению.

Модуль 1: Введение в программирование

Руководство по обучению

Ознакомление с модулем

Цель данного раздела – дать представление о навыках решения проблем, которые необходимы разработчикам программного обеспечения.

Модуль оценивает способность учащихся проектировать, создавать и тестировать компьютерную программу в соответствии с требованиями данного руководства. С самого начала объясните учащимся, что программирование в данном модуле основано на примере процедурного программирования. Хотя центры и вправе свободно выбирать любой процедурный язык высокого уровня, при условии, что включено содержание, указанное в спецификации, центрам рекомендуется использовать Python 3 при преподавании и оценке данного модуля.

Для изучения цели А начните с изучения концепций вычислительного мышления и их непосредственного отношения к навыкам решения проблем. Выполните это посредством обсуждения с целым классом и в малых группах о том, как эти концепции применяются в процессе решения проблем и компьютерного программирования. Дайте упражнения для выполнения в малых группах, чтобы изучить, как применять навыки решения проблем, определенные в спецификации, к данной концепции или проблеме.

Для цели обучения В важно дать ученикам практические занятия, которые позволят им развить свои навыки программирования. Изначально задания должны начинаться как короткие, автономные задачи, ориентированные на отдельные концепции или команды программирования. Со временем вы должны увеличить потребность в заданиях таким образом, чтобы учащиеся выбирали и объединяли ряд различных навыков.

Поощряйте учащихся делать заметки и создавать собственные справочные руководства, которыми они могут поделиться со своими сверстниками. Попросите учащихся создать общую папку для своих заметок на диске, в VLE центра или с помощью облачной службы, такой как Google Drive™ или Microsoft OneDrive®. Там, где позволяют задачи, т. е., когда они не являются оценочными, поощряйте учащихся работать совместно, предоставляя им обратную связь о своей работе, которую они провели, и при обсуждении решений.

Для цели обучения С вы должны убедиться, что учащиеся осведомлены о необходимости устойчивых к ошибкам и надежных программ и о роли тестирования. Поощряйте учащихся проводить тестирование документов и объясните, что итеративный подход к программированию приводит к гораздо более качественным результатам.

Вы должны давать учащимся соответствующую обратную связь на протяжении всего курса, предоставлять им возможность прочитать и понять полученную обратную связь, и побуждать их использовать инициативу для определения областей развития. Поощряйте учащихся работать над своими навыками программирования как можно больше вне классного времени. Программирование – это навык, который можно освоить эффективно, практикуя и повторяя.

Начало работы

Вначале приведен один из способов преподавания модуля.

Он основан на рекомендуемом методе оценивания, приведенном в спецификации.

Модуль 1: Введение в программирование

Введение

Надежно спроектированное и разработанное программное обеспечение может стать важной частью любого бизнеса или организации. Программное обеспечение, которое не работает или не полностью решает выявленную проблему, будет стоить денег и может подорвать репутацию бизнеса. Поэтому важно обеспечить, чтобы проблема была разбита на более мелкие, более управляемые части, и чтобы решение было подробно описано для разработки и полной проверки эффективной программы.

Цель обучения А – Составить план решения выявленной проблемы

- Рекомендуем начать с представления цели и того, как навыки решения проблем могут быть применены к практическим задачам.
- Объясните учащимся, что для решения проблем при разработке программного обеспечения используется «вычислительное мышление», которое позволяет нам анализировать проблему. Данный тип мышления имеет четыре стадии: декомпозиция, распознавание образов, обобщение и абстракция образов и разработка алгоритмов.
- Учащиеся работают в небольших группах, чтобы посмотреть на примере программы (такой как простая компьютерная игра) и определить, как можно применить каждый из четырех стадий вычислительного мышления. Например:
 - для «разбиения» учащиеся могут объяснить этапы игры, сценарий победы, роль конкретного персонажа в игре и т. д.;
 - для «распознавания образов» они могут определить, что некоторые неиграбельные персонажи двигаются одинаково;
 - для «обобщения и абстракции моделей» они могут определить требуемые переменные или начать определять требуемые входные данные;
 - для «разработки алгоритма» они могли бы описывать части игры, чтобы планировать логические шаги, необходимые для готовности к написанию кода, например, описать, что происходит, когда играбельный персонаж вступает в контакт с определенным элементом.

Модуль 1: Введение в программирование

- Тема А2 определяет знания, навыки и понимание, необходимые для разработки алгоритма в теме А1. о этой теме учащихся не требуется ознакомлять с программным кодом. На данном этапе, однако, должны быть введены некоторые понятия, такие как «последовательность, отбор и итерация», например, для определения чего-то, что требует повторения, почему это так и какую форму примет это повторение.
- Дайте учащимся возможность изучить ряд проблем, к которым они могут применить свои навыки решения проблем.

Цель обучения В – Создать программный код для решения выявленной проблемы

- Данная цель обучения даст учащимся инструменты, необходимые для создания программы. При рассмотрении этой цели обучения убедитесь, что вы выделяете пример процедурного программирования. Рекомендуется использовать центры Python 3 для преподавания и оценки данного модуля.
- Преподавание содержания данной учебной цели должно быть как можно более практическим. Начните с коротких автономных задач, которые сосредоточены на отдельных концепциях программирования или командах. Со временем потребность в занятиях возрастает таким образом, что учащиеся должны выбрать и объединить ряд различных навыков.
- Когда вы уверены, что учащиеся понимают концепцию, дайте им несколько практических упражнений, чтобы попрактиковаться в работе с концепцией. Например, вводя концепцию «выбора», дайте учащимся ряд различных сценариев, для которых они должны разрабатывать небольшие фрагменты кода. Ученикам может быть предоставлена рабочая тетрадь, в которой содержится ряд задач, которые им предстоит выполнить, и недостатков, которые необходимо исправить.
- По мере того, как учащиеся развиваются свои навыки программирования и выполняют различные задачи, было бы полезно периодически собирать их вместе, чтобы повторить уже освоенные навыки. Кроме того, когда упражнение оказывается сложным для некоторых учащихся, вы можете работать с модельными ответами в качестве целого упражнения. Это также даст вам возможность задать учащимся прямые вопросы, чтобы проверить их понимание темы. Вы можете попросить более сильных программистов в классе сотрудничать или работать с менее способными программистами.

Модуль 1: Введение в программирование

- На регулярной основе в течение всего времени преподавания модуля вы должны давать короткие задания на оценку. Данные задания следует использовать для оценки успеваемости и определения слабых сторон, которые необходимо подтянуть.
- После того как весь материал пройден, вы должны выставить ученикам пробную оценку. Пробная оценка должна отражать внешнюю оценку, которую они получат, поэтому она должна предоставить соответствующую проблему, для которой они могут создать решения. Эта задача должна быть выполнена независимо в контролируемых условиях, которые повторяют условия финальной оценки.

Цель обучения С – Протестировать и доработать компьютерную программу

- Данную цель обучения лучше всего преподавать в сочетании с целью обучения В. После того как учащиеся освоят некоторые базовые навыки программирования, объясните важность постоянного тестирования.
- Изучите вместе с учащимися идею о том, что из-за размера большинства проектов разработки программного обеспечения программисты редко работают самостоятельно и часто являются частью гораздо большей команды разработчиков. Объясните, что в больших программах ошибки неизбежны. Поэтому важно записывать процесс тестирования, чтобы проблемы можно было отслеживать и устранять быстрее. План тестирования или журнал испытаний также являются важным способом передачи требований для тестирования большей команде.
- Исследуйте вместе с учащимися важность надежных программ и ту роль, которую тестирование играет в обеспечении надежности программ.

Подробности ссылок на другие модули ВТЕС и квалификации

Данный модуль эффективно преподавать одновременно с Модулем 2: Анализ и дизайн программного обеспечения

Ресурсы

Веб-сайты

<https://docs.python.org/3/>

Официальный сайт документации по языку программирования Python.

www.learnpython.org/

Интерактивное руководство по программированию на Python. Полезно для введения отдельных команд или в качестве справочного руководства.

Учащиеся также могут использовать его для работы над своими навыками программирования вне классных занятий.

www.w3schools.com/python/

Интерактивное руководство по программированию на Python. Полезно для введения отдельных команд или в качестве справочного руководства, учащиеся могут использовать его для работы над своими навыками программирования вне классных занятий. Сайт также предоставляет учебные пособия для других языков программирования, таких как HTML и JavaScript, которые могут быть полезны для модуля 3.

Pearson не несет ответственности за содержание любых внешних интернет-сайтов. Для преподавателей важно предварительно просмотреть каждый веб-сайт, прежде чем использовать его в классе, чтобы убедиться, что URL-адрес по-прежнему точен, актуален и целесообразен. Мы рекомендуем преподавателям добавлять полезные веб-сайты в закладки и предлагать учащимся доступ к ним через интранет-сеть школы / колледжа.

Схемы работы

Модуль	Модуль 1: Введение в программирование
Учебные часы	90
Количество уроков	45
Продолжительность уроков	2 часа
Ссылки на другие модули	Модуль 2: Анализ программного обеспечения и дизайн

Ключи к способу обучения			
ОУ	Оценочное упражнение	ЗМ	Закрепление материала
ПС	Приглашенный спикер	В	Визит
СО	Самостоятельное обучение	ОР	Опыт работы

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
1	A1 Навыки решения проблем: <ul style="list-style-type: none">• Разбиение.	СО	<ul style="list-style-type: none"> • Начальное упражнение: введение в модуль и основные темы, которые будут освещены. Объясните важность навыков решения проблем в программировании. Объясните, что решение проблемы будет применяться на протяжении всей квалификации. • Демонстрация под руководством преподавателя: объясните разбиение на части. Покажите разбиение на части на примере компьютерной проблемы. • Упражнение в малых группах / парах: учащиеся практикуют разбиение на части по заданному сценарию. • Обсуждение под руководством преподавателя: учащиеся обмениваются идеями со всей группой. • Заключительное упражнение: дать учащимся возможность улучшить навыки разбиения, основываясь на обсуждении. 	<ul style="list-style-type: none"> • Характеристики модуля. • Флипчарт или аналог для учащихся для записи обсуждений и идей. • Презентация. • Пример компьютерной проблемы. • Пример проблемного сценария для учащихся.

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
2	<p>A1 Навыки решения проблем:</p> <ul style="list-style-type: none"> • Распознавание шаблона. • Обобщение и абстракция шаблона. 	СО	<ul style="list-style-type: none"> • Начальное упражнение: перескажите концепцию разбиения в сессии вопросов и ответов. Объясните, что в данном уроке учащиеся продолжат изучать навыки решения проблем. • Демонстрация под руководством преподавателя: объяснение распознавания шаблонов, обобщения и абстракции шаблона. Покажите на примере компьютерной проблемы. • Упражнение в малых группах/парах: учащиеся рассматривают данную проблему и анализируют ее с помощью распознавания, обобщения и абстракции шаблонов. • Обсуждение под руководством преподавателя: попросите учеников поделиться своими мыслями. • Заключительное упражнение: используя то, что они узнали из обсуждения, учащиеся завершают работу над сценарием упражнения из предыдущего урока. 	<ul style="list-style-type: none"> • Характеристики модуля. • Флипчарт или аналог для учащихся для записи обсуждений и идей. • Презентация. • Работа учащихся с прошлого урока. • Пример сценария проблемы с прошлого урока.

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
3	<p>A1 Навыки решения проблем:</p> <ul style="list-style-type: none"> • Дизайн алгоритма. <p>A2 Техники, используемые для разработки алгоритмов.</p>	СО	<ul style="list-style-type: none"> • Начальное упражнение: введение в урок, объяснение терминов <i>последовательность, выбор и итерация</i>. Объясните, что в течение следующих двух уроков учащиеся познакомятся с проектированием алгоритма. • Обсуждение под руководством преподавателя: используйте вопросы для проверки понимания учащимися реляционных операторов. • Упражнение в малых группах: дайте учащимся возможность обсудить проблему. Учащиеся должны определить порядок выполнения задач. • Демонстрация под руководством преподавателя: представить идею выбора (ветвления), включая операторы If... Then... Else и Булевы. • Задача для малой группы: дать ученикам задачу для обсуждения. Учащиеся определяют выбор и итерацию, которая произойдет. • Итоговое упражнение: повторите основные пункты урока. 	<ul style="list-style-type: none"> • Характеристики модуля. • Сценарий и работы учащихся с 1 и 2 уроков. • Презентация.

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
4-5	<p>A1 Навыки решения проблем:</p> <ul style="list-style-type: none"> • Дизайн алгоритма. <p>A2 Техники, используемые для разработки алгоритмов.</p>	СО	<ul style="list-style-type: none"> • Начальное упражнение: повторите концепции разбиения, распознавания, обобщения и абстракции шаблонов с помощью сессии вопросов и ответов. • Упражнение в малых группах/парах: поставьте перед учащимися проблему и попросите их рассмотреть «алгоритмы» для разных частей проблемы. • Упражнение в группах (взаимная проверка и отзывы): на разных этапах в течение двух уроков попросите группы обсудить свою работу с другими небольшими группами. • Заключительное упражнение: группы комментируют свою работу во время обсуждений, чтобы определить любые изменения, которые будут внесены в результат обратной связи с однокурсниками. 	<ul style="list-style-type: none"> • Параметры модуля. • Листы бумаги крупного формата. • Работа учащихся с прошлого урока. • Пример проблемного сценария для учащихся. • Цветные ручки (для примечаний).

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
6	A1 Навыки решения проблем.	ОУ	<ul style="list-style-type: none"> • Начальное упражнение: организуйте учащихся так, чтобы они могли самостоятельно выполнять задания на оценку. • Самостоятельное оценочное упражнение: дайте учащимся сценарий. Они должны составить план решения проблемы. Разработанный план должен охватывать все основные пункты содержания в учебной цели А. <p>Примечание: преподавателям преподаватели должны использовать результаты оценочного упражнения для мониторинга успеваемости учащихся. Данная оценка не является итоговой оценкой.</p>	<ul style="list-style-type: none"> • Параметры модуля. • Оценочное упражнение, содержащее краткое описание и сценарий. • Копии содержания для учебной цели А (справочный лист навыков решения проблем).

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
7	Цель обучения В <ul style="list-style-type: none"> Обзор процедурного программирования Последовательность В3. 	СО	<ul style="list-style-type: none"> Начальное упражнение: объясните учащимся, что теперь они будут рассматривать конкретный пример программирования (процедурное программирование). Презентация преподавателя: обзор особенностей и характеристик примера процедурного программирования. Ознакомление учащихся со структурой процедурного программирования. Самостоятельное упражнение: учащиеся должны изучить и сделать заметки об использовании, преимуществах и недостатках процедурного программирования. Упражнение в малых группах: организуйте учащихся в небольшие группы, чтобы обсудить полученные ими результаты. Позвольте им добавлять и расширять свои заметки, основываясь на этих обсуждениях. Итоговое упражнение: учащиеся обсуждают изученные темы. 	<ul style="list-style-type: none"> Параметры модуля. Презентация. Компьютеры с доступом в интернет. Учебники.

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
8	<p>В1 Обработка данных в программе:</p> <ul style="list-style-type: none"> Константы и переменные. <p>В2 Арифметические операции:</p> <ul style="list-style-type: none"> Математические операторы. <p>В4 Встроенные функции и библиотеки:</p> <ul style="list-style-type: none"> Принятие ввода пользователя. Вывод сообщений на экран. Преобразование числовых значений в строки. Преобразование строк в числовые значения. 	СО	<ul style="list-style-type: none"> Начальное упражнение: объясните, что на следующем этапе занятий учащиеся знакомятся с основами программирования. Презентация преподавателя: представьте базовую терминологию, которая будет использоваться на этом занятии, и среду разработки, которая будет использоваться. Самостоятельное упражнение: учащиеся выполняют ряд базовых задач программирования (принимая вводимые пользователем данные, присваивая значение переменной, выполняя простые математические уравнения). Итоговое упражнение: учащиеся обсуждают свой опыт написания кода и выполняют тестирование по темам, затронутым в данном уроке. 	<ul style="list-style-type: none"> Параметры модуля. Презентация. Список упражнений по программированию. Компьютеры с интегрированной средой разработки (IDE). Официальная документация по Python и онлайн-уроки программирования. [Примечание: данные ресурсы будут полезны во всех следующих уроках, кроме упражнений на оценку]. Контрольный опрос в рамках итогового упражнения.

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
9	<p>В1 Обработка данных в программе:</p> <ul style="list-style-type: none"> Список / массив. <p>В2 Арифметические операции:</p> <ul style="list-style-type: none"> Реляционные операторы. Булевые операторы. <p>В3 Управляющие структуры:</p> <ul style="list-style-type: none"> Выбор (ветвление). 	СО	<ul style="list-style-type: none"> Начальное упражнение: вспомните работу, выполненную в предыдущем уроке, и проверьте понимание учащихся с помощью примеров кода. Обсуждение под руководством преподавателя: исследуйте концепцию необходимости хранить несколько значений (списков элементов) и необходимости разрешать ветвление программ (обращаясь к уроку 3 по мере необходимости). Демонстрация под руководством преподавателя: продемонстрируйте, как создать список и как создавать операторы IF с использованием Python. Самостоятельное упражнение: учащиеся выполняют ряд задач программирования в списках и выражениях IF. Итоговое упражнение: учащиеся работают в парах, чтобы посмотреть код друг друга и предложить улучшения. 	<ul style="list-style-type: none"> Параметры модуля. Примеры кодов. Презентация. Компьютеры с интегрированной средой разработки (IDE). Список упражнений по программированию.

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
10	<p>В3 Управляющие структуры:</p> <ul style="list-style-type: none"> Итерация. <p>В4 Встроенные функции и библиотеки:</p> <ul style="list-style-type: none"> Определение диапазона. <p>В6 Принципы разработки качественного кода:</p> <ul style="list-style-type: none"> Легкий в сопровождении код (комментарии к коду). 	СО	<ul style="list-style-type: none"> Начальное упражнение: вспомните работу, выполненную в предыдущем уроке, и проверьте понимание учащихся с помощью примеров кода. Обсуждение под руководством преподавателя: исследуйте концепцию необходимости повторения процессов в программе (по необходимости, возвращаясь к уроку 3). Демонстрация под руководством преподавателя: покажите, как создать цикл «For» и цикл «While» с помощью Python. Самостоятельное упражнение: учащиеся выполняют ряд задач по программированию на итерации. Итоговые упражнения: продемонстрировать, как комментировать код (комментарии к коду). Учащиеся комментируют свой код, чтобы объяснить, как он работает. 	<ul style="list-style-type: none"> Параметры модуля. Компьютеры с интегрированной средой разработки (IDE). Примеры кодов. Презентация. Список упражнений по программированию.

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
11	Цель обучения В.	ОУ	<ul style="list-style-type: none"> • Начальное упражнение: организуйте учащихся так, чтобы они могли самостоятельно выполнять задания на оценку. • Самостоятельное оценочное упражнение: дайте учащимся ряд задач по программированию. Задания должны быть по материалам 8-10 уроков. <p>Примечание: преподаватели должны использовать результаты оценочного упражнения для мониторинга успеваемости учащихся. Данная оценка не является итоговой оценкой.</p>	<ul style="list-style-type: none"> • Параметры модуля. • Оценочное упражнение. • Компьютеры с интегрированной средой разработки (IDE).

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
12	<p>В4 Встроенные функции и библиотеки:</p> <ul style="list-style-type: none"> Генерация случайных чисел. 	СО	<ul style="list-style-type: none"> Начальное упражнение: повторение 10 урока. Объясните, что в этом уроке мы рассмотрим некоторые из встроенных функций, доступных в Python. Обсуждение под руководством преподавателя: исследуйте концепцию необходимости создания программ гибкими и непредсказуемыми. Демонстрация под руководством преподавателя: продемонстрируйте, как генерировать случайные числа с помощью Python. Индивидуальное упражнение: учащиеся прорабатывают ряд заданий и программных заданий по использованию случайных чисел. Итоговое упражнение: в парах учащиеся обсуждают свою работу и обмениваются отзывами с однокурсниками. 	<ul style="list-style-type: none"> Параметры модуля. Презентация. Компьютеры с интегрированной средой разработки (IDE). Список упражнений и задач по программированию.

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
13	<p>В1 Обработка данных в программе:</p> <ul style="list-style-type: none"> • Применение локальных и глобальных переменных. • Соглашения. <p>В6 Принципы разработки качественного кода:</p> <ul style="list-style-type: none"> • Создание читаемого кода: имена переменных; соглашения об именах; интервал. Создание легкого в сопровождении кода: встроенные библиотеки; модульная конструкция – использование автономных функций. 	СО	<ul style="list-style-type: none"> • Начальное упражнение: вспомните предыдущий урок и напомните учащимся об использовании встроенных функций. Изучите назначение функций и необходимость иногда создавать свои собственные функции. • Демонстрация под руководством преподавателя: продемонстрируйте, как генерировать, определять и описывать функцию и как использовать локальные и глобальные переменные. • Самостоятельное упражнение: учащиеся прорабатывают ряд заданий и программируют задачи по созданию функций. • Итоговое упражнение: в парах учащиеся обсуждают свою работу и обмениваются отзывами с одногруппниками. 	<ul style="list-style-type: none"> • Параметры модуля. • Презентация. • Список упражнений по программированию. • Компьютеры с интегрированной средой разработки (IDE).

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
14	<p>В4 Встроенные функции и библиотеки:</p> <ul style="list-style-type: none"> • Округление числа до определенного количества десятичных знаков. • Усечение числа. • Поиск самого большего или самого меньшего значения в группе чисел. 	СО	<ul style="list-style-type: none"> • Начальное упражнение: вспомните предыдущий урок и объясните другие встроенные функции. Вспомните использование списков. • Демонстрация под руководством преподавателя: продемонстрируйте использование функций округления, усечения, минимума и максимума в Python. • Самостоятельное упражнение: учащиеся выполняют ряд программных заданий по созданию функций • Итоговое упражнение: сессия вопросов и ответов по темам, затронутым в этом уроке. 	<ul style="list-style-type: none"> • Параметры модуля. • Презентация. • Компьютеры с интегрированной средой разработки (IDE). • Список упражнений по программированию.

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
15	<p>В4 Встроенные функции и библиотеки:</p> <ul style="list-style-type: none"> Проверить, написана ли строка в верхнем или нижнем регистре. Проверить, содержит ли строка числовые значения. <p>Преобразовать строку из верхнего регистра в нижний регистр.</p> <ul style="list-style-type: none"> Преобразовать строку из нижнего регистра в верхний регистр. 	СО	<ul style="list-style-type: none"> Начальное упражнение: вспомните 8 урок о преобразовании строки в число. Также кратко повторите 14 урок. Объясните, что на этом уроке мы продолжим изучать больше встроенных функций. Обсуждение под руководством преподавателя: изучите необходимость сделать код устойчивым, включая проверку пользовательского ввода. Демонстрация под руководством преподавателя: продемонстрируйте использование встроенных функций в Python для проверки текста / числа и преобразования между строчными и прописными буквами Самостоятельное упражнение: учащиеся выполняют ряд программных заданий по созданию функций. Итоговое упражнение: учащиеся выполняют тест по темам, которые рассматриваются в этом уроке. 	<ul style="list-style-type: none"> Параметры модуля. Презентация. Компьютеры с интегрированной средой разработки (IDE). Список упражнений по программированию. Контрольный опрос в рамках итогового упражнения.

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
16	<p>B1 Обработка данных в программе:</p> <ul style="list-style-type: none"> • Список / массив. • Словарь/карта. <p>B4 Встроенные функции и библиотеки:</p> <ul style="list-style-type: none"> • Использовать функции для обработки данных в списке / массиве (добавление данных; вставка данных в определенное расположение; удаление данных; всплывающие данные). 	СО	<ul style="list-style-type: none"> • Начальное упражнение: вспомните предыдущий урок. Используйте сессию вопросов и ответов, чтобы подвести итоги и проверить понимание учащимися базового использования списков / массивов (при необходимости обращайтесь к уроку 9). • Демонстрация под руководством преподавателя: продемонстрируйте использование списков / массивов (включая индексацию). • Самостоятельное упражнение: учащиеся выполняют ряд задач программирования, используя списки / массивы. • Итоговое упражнение: учащиеся работают в парах, чтобы обсудить свой код и предоставить обратную связь. 	<ul style="list-style-type: none"> • Параметры модуля. • Презентация. • Компьютеры с интегрированной средой разработки (IDE). • Список упражнений по программированию.

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
17	<p>В1 Обработка данных в программе:</p> <ul style="list-style-type: none"> Список / массив. Словарь/карта. <p>В4 Встроенные функции и библиотеки:</p> <ul style="list-style-type: none"> Используют функции для обработки данных в списке / массиве (проверяют, находится ли определенный символ, слово или значение в списке / массиве; возвращают местоположение определенного символа, слова или значения; возвращают количество раз, когда определенный символ, слово или значение находится в списке / массиве; сортируют элементы в списке / массиве в порядке возрастания или убывания). 	СО	<ul style="list-style-type: none"> Начальное упражнение: вспомните предыдущий урок. Используйте сессию вопросов и ответов, чтобы подвести итоги и проверить понимание учащихся. Обсуждение под руководством преподавателя: выясните, почему в программах нужно хранить несколько значений. Обсудите некоторые варианты использования списков. Демонстрация под руководством преподавателя: продемонстрируйте использование различных функций для управления списками / массивами. Самостоятельное упражнение: учащиеся выполняют ряд задач программирования, используя функции, которые были продемонстрированы. Демонстрация под руководством преподавателя: продемонстрируйте использование словаря / карты. Выясните, чем он отличается от списка / массива. Самостоятельное упражнение: учащиеся выполняют ряд задач программирования, используя словари / карты. Итоговое упражнение: учащиеся работают в парах, чтобы обсудить свой код и предоставить обратную связь друг другу. 	<ul style="list-style-type: none"> Параметры модуля. Презентации. Компьютеры с интегрированной средой разработки (IDE). Список упражнений по программированию.

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
18	<p>В4 Встроенные функции и библиотеки:</p> <ul style="list-style-type: none"> Разделяют строку на отдельные символы. Проверяют, находится ли конкретная буква или слово в строке. Выводят местоположение определенного символа в строке. Проверяют длину строки. 	СО	<ul style="list-style-type: none"> Начальное упражнение: подведение итогов (урок 16). Демонстрация под руководством преподавателя: продемонстрируйте использование индексации для поиска определенных букв в строке и других функций обработки строк (включая <code>split()</code>, <code>len()</code>, <code>in</code>). Самостоятельное упражнение: учащиеся выполняют ряд задач программирования, используя функции, которые были продемонстрированы. Итоговые упражнения: учащиеся обмениваются одним из своих фрагментов кода на сегодняшнем уроке с другим учеником. Учащиеся пытаются вызвать сбой программы, а затем записывают выполненные действия и результат. 	<ul style="list-style-type: none"> Параметры модуля. Презентация. Компьютеры с интегрированной средой разработки (IDE). Список упражнений по программированию.

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
19	<p>В5 Проверка данных:</p> <ul style="list-style-type: none"> Используйте встроенные или пользовательские функции для применения методов проверки к программному коду. Напишите код, который может обрабатывать некорректный ввод без ошибок. Используйте итерацию, чтобы программа не продолжалась до тех пор, пока не будет получен приемлемый ввод. Генерируйте выходные данные программы, которые предоставляют пользователю полезные сообщения о любых ошибках ввода данных и о том, как их исправить. 	СО	<ul style="list-style-type: none"> Начальное упражнение: подытожите заключительное упражнение из урока 18. Обсуждение под руководством преподавателя: изучите необходимость обеспечения надежности программ. <p>Обсудите конкретные сценарии, в которых программы должны будут проверять пользовательский ввод</p> <ul style="list-style-type: none"> Демонстрация под руководством преподавателя: продемонстрируйте использование соглашений в программировании для проведения проверки в программах. Индивидуальные упражнения: учащиеся прорабатывают ряд задач по программированию и решают задачу программирования на проверку. Итоговое упражнение: учащиеся меняют один из своих фрагментов кода на текущем уроке на фрагмент кода другого учащегося. Учащиеся пытаются вызвать сбой программы и предоставляют обратную связь. 	<ul style="list-style-type: none"> Параметры модуля. Презентация. Компьютеры с интегрированной средой разработки (IDE). Список упражнений по программированию. Пример требований к коду. Справочные листы кода (включая циклы, обработку строк, индексацию).

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
20	<p>В4 Встроенные функции и библиотеки:</p> <ul style="list-style-type: none"> • Открывать внешние файлы. • Читать из внешнего файла. • Записывать во внешний файл. 	СО	<ul style="list-style-type: none"> • Начальное упражнение: вспомните использование структур данных (массив / список, словарь). Объясните, что иногда более эффективно хранить данные вне программного кода. • Демонстрация под руководством преподавателя: продемонстрируйте использование встроенных функций для манипулирования внешним файлом .txt. • Самостоятельное упражнение: учащиеся выполняют серию задач по программированию внешних файлов .txt. • Итоговое упражнение: учащиеся проходят тестирование по темам, которые они затронули на уроке. 	<ul style="list-style-type: none"> • Параметры модуля. • Презентация. • Компьютеры с интегрированной средой разработки (IDE). • Список упражнений по программированию. • Контрольный опрос в рамках общего задания.
21	Цель обучения В.	ОУ	<ul style="list-style-type: none"> • Начальное упражнение: организуйте учащихся так, чтобы они могли самостоятельно выполнять задания на оценку. • Самостоятельное оценочное упражнение: дайте учащимся ряд задач по программированию. Задания должны быть по материалам 8-19 уроков. <p>Примечание: преподавателям должны использовать результаты оценочного упражнения для мониторинга успеваемости учащихся. Данная оценка не является итоговой оценкой.</p>	<ul style="list-style-type: none"> • Параметры модуля. • Оценочное упражнение. • Компьютеры с интегрированной средой разработки (IDE). • Справочный лист по кодированию Python или справочник по кодам и синтаксису.

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
22	<p>C1 Написание плана тестирования:</p> <ul style="list-style-type: none"> Цель идентифицированного теста. Использование данных теста. Предполагаемые результаты теста для выбранных тестов. 	СО	<ul style="list-style-type: none"> Начальное упражнение: подытожим ключевые моменты из уроков 18 и 19 о надежных программах. Обсуждение под руководством преподавателя: исследуйте важность надежных программ и роль, которую тестирование играет в обеспечении надежности программ. Демонстрация под руководством преподавателя: объясните особенности плана тестирования и различные типы данных теста. Объясните, почему для разработчика программного обеспечения важно документировать свое тестирование. Самостоятельное упражнение: учащиеся создают подробный план тестирования для данного фрагмента кода. Итоговые упражнения: учащиеся в парах обсуждают свои планы тестирования. Они должны быть готовы объяснить или обосновать свой выбор тестов и данных испытаний. Они также могут добавить дополнительные тесты в свой план тестирования. 	<ul style="list-style-type: none"> Параметры модуля. Презентация. Компьютеры с интегрированной средой разработки (IDE). Частично заполненный программный код и программные требования. Шаблон плана испытаний.

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
23	<p>C2 Разработка итеративного кода:</p> <ul style="list-style-type: none"> Систематически применять план тестирования и связанные с ним тестовые данные. 	СО	<ul style="list-style-type: none"> Начальное упражнение: вспомните ключевые моменты из предыдущего урока. Демонстрация под руководством преподавателя: демонстрация процесса применения запланированных тестов, записи результатов, исправления кода и повторного тестирования. Самостоятельное упражнение: учащиеся применяют свой план тестирования к данному коду для выявления ошибок и улучшения кода по мере необходимости. Итоговое упражнение: учащиеся комментируют код, чтобы определить, как он работает и какие изменения они внесли. 	<ul style="list-style-type: none"> Параметры модуля. Презентация. Компьютеры с интегрированной средой разработки (IDE). Частично заполненный программный код и программные требования из предыдущего урока. Планы тестирования учащихся из предыдущего урока.

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
24	Учебная цель А.	СО	<ul style="list-style-type: none"> • Начальное упражнение: объясните учащимся, что на следующих уроках они выполнят задание, которое позволит им практиковать все, чему они научились в данном разделе. • Обсуждение под руководством преподавателя: подытожить ключевые моменты цели обучения А (уроки 1–6). Объясните, что учащиеся получат задачу, для которой требуется компьютерная программа. Учащиеся планируют, создают и тестируют программу. • Самостоятельное упражнение: учащиеся составляют подробный план программы (при необходимости обращаясь к цели обучения А). • Заключительное упражнение: парами учащиеся обсуждают свои планы. Они должны быть готовы объяснить или обосновать свой выбор. Затем учащиеся могут предоставить обратную связь одногруппникам. 	<ul style="list-style-type: none"> • Параметры модуля. • Презентация. • Компьютеры с интегрированной средой разработки (IDE). • Руководство по решению задач. • Копии содержания для учебной цели А (справочный лист навыков решения проблем).

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
25	Цели обучения В и С.	СО	<ul style="list-style-type: none"> • Начальное упражнение: повторите предыдущий урок. • Обсуждение под руководством преподавателя: объясните важность итеративной разработки и необходимость тестирования в течение всей разработки программы. Объясните, что тестирование должно быть запланировано до начала разработки, а затем добавлено к нему по мере развития программы. • Самостоятельное упражнение: учащиеся создают план тестирования для программы, которую они будут создавать. • Упражнение в малых группах/парах: в парах учащиеся обсуждают свои планы. Они должны быть готовы объяснить или обосновать свой выбор и предоставить обратную связь. • Самостоятельное упражнение: учащиеся начинают создавать программу на основе своих планов, используя материалы этого и предыдущего уроков. 	<ul style="list-style-type: none"> • Параметры модуля. • Работа учащихся с прошлого урока. • Презентация. • Руководство по решению задач. • Шаблон плана испытаний. • Компьютеры с интегрированной средой разработки (IDE).

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
26 - 27	Цели обучения В и С.	ОУ	<ul style="list-style-type: none"> • Начальное упражнение: повторение пройденных уроков. • Индивидуальное оценочное упражнение: учащиеся разрабатывают и тестируют свои программы в соответствии с заданием. Используйте результаты уроков 24–27, чтобы определить сильные стороны учащихся и области для улучшения. <p>Примечание: Данная оценка не является окончательной оценкой.</p>	<ul style="list-style-type: none"> • Параметры модуля. • Работа учащихся с прошлого урока. • Руководство по решению задач. • Шаблон плана испытаний. • Справочник по языку и синтаксису. • Компьютеры с интегрированной средой разработки (IDE).

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
28-29	Цели обучения А, В и С.	ЗМ	<ul style="list-style-type: none"> • Начальное упражнение: предоставьте учащимся свой отзыв о работе, выполненной в уроках 24–27. • Самостоятельное упражнение: учащиеся читают отзывы, которые им дали, и определяют области, которые требуют дальнейшего изучения или улучшения. • Самостоятельная работа / работа в парах: учащиеся выполняют индивидуальное или парное задания в областях, которые им необходимо улучшить. 	<ul style="list-style-type: none"> • Параметры модуля. • Отзывы преподавателя о работе по урокам 24–27. • Работа учащихся по урокам 24–27. • Практические упражнения по программированию. • Компьютеры с интегрированной средой разработки (IDE).

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
30 - 32	Цели обучения А, В и С.	ОУ	<ul style="list-style-type: none"> • Начальное упражнение: вспомните процесс, которому учащиеся следовали в уроках 24–27. Организуйте учащихся так, чтобы они могли самостоятельно выполнять задания на оценку. • Самостоятельное оценочное упражнение: дайте учащимся задачу, для выполнения которой они должны спланировать, создать и протестировать решение. • Самостоятельное обучение: учащиеся могут посвятить внеклассное время выполнению некоторых задач. <p>Примечание: преподавателям преподаватели должны использовать результаты оценочного упражнения для мониторинга успеваемости учащихся. Данная оценка не является итоговой оценкой.</p>	<ul style="list-style-type: none"> • Параметры модуля. • Руководство по решению задач. • Шаблон плана испытаний. • Справочник по языку и синтаксису. • Компьютеры с интегрированной средой разработки (IDE).

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
33 - 34	Цели обучения А, В и С.	ЗМ	<ul style="list-style-type: none"> • Начальное упражнение: предоставьте учащимся копии отзывов преподавателя о работе, проведённой в уроках 30–32. • Самостоятельное упражнение: учащиеся читают отзывы и определяют области, которые требуют дальнейшего изучения или улучшения. • Самостоятельная работа / работа в парах: учащиеся работают самостоятельно или в парах в тех областях, которые им необходимо улучшить. 	<ul style="list-style-type: none"> • Параметры модуля. • Отзывы учителей о работе по урокам 30-32. • Практические упражнения по программированию. • Компьютеры с интегрированной средой разработки (IDE).
35 - 39	Цели обучения А, В и С. Практический Практическая самостоятельная работа.	ОУ	<ul style="list-style-type: none"> • Начальное упражнение: объясните, что в следующих пяти уроках учащиеся выполняют Практическая самостоятельная работа для подготовки к итоговой контрольной. • Самостоятельное оценочное упражнение: учащиеся выполняют Практическая самостоятельная работа, схожий с итоговой контрольной. <p>Примечание: используемое оценочное упражнение должно соответствовать структуре, аналогичной итоговой оценке, и должно оцениваться по тем же критериям. Однако оно не должно быть тем же самым, что будет использоваться в итоговой контрольной по модулю.</p>	<ul style="list-style-type: none"> • Практическое оценочное упражнение. • Шаблон плана испытаний. • Справочник по языку и синтаксису. • Компьютеры с интегрированной средой разработки (IDE).

№	Тема	Тип урока	Предлагаемые упражнения	Ресурсы
40	Цели обучения А, В и С.	ЗМ	<ul style="list-style-type: none"> • Начальное упражнение: дать учащимся их оцененную работу и отзывы о результатах практической самостоятельной работы (уроки 35–39). • Самостоятельное упражнение: ознакомление учащихся с проверенной работой и работа над ошибками. • Самостоятельная работа / работа в парах: учащиеся выполняют индивидуальное или парное задание по пунктам, которые им необходимо улучшить. 	<ul style="list-style-type: none"> • Параметры модуля. • Работа учащихся по урокам 35–39. • Отзывы преподавателя о практических самостоятельных работах из уроков 35–39. • Практические упражнения по программированию. • Компьютеры с интегрированной средой разработки (IDE).
41 – 45	Цели обучения А, В и С. Итоговая оценка.	ОУ	<ul style="list-style-type: none"> • Начальное упражнение: представьте самостоятельную работу (см. «Рабочая тетрадь для оценки»), подробно описав основные критерии оценки, сроки и критерии представления. • Самостоятельное оценочное упражнение: учащиеся используют «Рабочую тетрадь для оценки» для завершения модуля 1: Задания 1, 2 и 3. 	<ul style="list-style-type: none"> • Параметры модуля. • Рабочая тетрадь для оценки. • Компьютеры с интегрированной средой разработки (IDE). • Шаблон плана испытаний.

План урока

Квалификация	Pearson BTEC Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	1 (2 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none"> • понимать необходимость применения навыков решения проблем для их анализа. • уметь идентифицировать и описать проблему. • уметь разбивать проблему на более мелкие части. • уметь описать основные особенности проблемы или процесса.
-------------------	---

Список ресурсов	<ul style="list-style-type: none"> • Параметры модуля. • Флипчарт или аналог для учащихся для записи обсуждений и идей. • Презентация. • Пример проблемного сценария для учащихся.
------------------------	--

Ключ: **СУ**: Список упражнений; **ШФ**: шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (15 минут)	<ul style="list-style-type: none"> Представьте блок и основные темы, которые будут рассмотрены. Объясните важность навыков решения проблем в программировании. Объясните, что учащиеся должны будут применять навыки решения проблем во всех модулях, которые они изучают на протяжении данной квалификации.
Демонстрация под руководством преподавателя (20 минут)	<ul style="list-style-type: none"> Покажите презентацию, чтобы объяснить разбиение. Продемонстрируйте, как применяется разбиение на примере компьютерной проблемы. Например, это может быть компьютерная программа, используемая гостиницей для управления бронированием номеров. Разбиение должно акцентировать внимание на: <ul style="list-style-type: none"> выявлении и описании проблемы разбиении проблемы на более мелкие части описании основных особенностей проблемы или процесса. Используйте метод вопросов и ответов на протяжении всего урока, чтобы участвовать в работе учащихся с примером по разбиению.
Упражнение в малых группах/парах (45 минут)	<ul style="list-style-type: none"> Учащиеся практикуют разбиение, используя следующий сценарий. <ul style="list-style-type: none"> Вас попросили создать компьютерную игру, подобную классической игре Tetris®. Разбейте проблему. Убедитесь, что вы: <ul style="list-style-type: none"> - определили и описали проблему - разбили проблему на более мелкие части - описали основные особенности проблемы или процесса. Учащиеся записывают свои ответы на компьютере или на бумаге.

Упражнения	Учебные заметки
Обсуждение под руководством преподавателя (20 минут)	<ul style="list-style-type: none">• Дайте учащимся возможность оставить отзыв о работе остальным участникам группы.• Позвольте ученикам обсудить, как их работа отличается от работы других учеников.
Заключительное упражнение (20 минут)	<ul style="list-style-type: none">• Дайте учащимся возможность отработать навык разбиения. Учащиеся должны использовать идеи, которые они разработали в ходе обсуждения, для дополнения или усовершенствования своей работы.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	2 (2 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none"> • понимать необходимость применения навыков решения проблем для анализа проблем. • уметь определять общие элементы или особенности процессов, или проблем. • уметь определять различия между процессами или проблемами. • уметь идентифицировать отдельные элементы внутри процессов или проблем. • быть в состоянии описать шаблоны, которые были определены. • уметь делать прогнозы на основе выявленных закономерностей.
-------------------	---

Список ресурсов	<ul style="list-style-type: none"> • Параметры модуля. • Флипчарт или аналог для учащихся для записи обсуждений и идей. • Презентация. • Работа учащихся с прошлого урока. • Пример сценария проблемы с прошлого урока.
------------------------	--

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (10 минут)	<ul style="list-style-type: none"> Повторите концепцию разбиения, используя сессию вопросов и ответов. Объясните, что на этом уроке учащиеся продолжат изучать навыки решения проблем, в частности: <ul style="list-style-type: none"> распознавание шаблона. обобщение образца. абстракция.
Демонстрация под руководством преподавателя (20 минут)	<ul style="list-style-type: none"> Проведите презентацию, чтобы объяснить распознавание образов, обобщение и абстракцию. Продемонстрируйте, как эти навыки применяются на примере компьютерной проблемы. Используйте тот же пример, который вы показывали в демонстрации на уроке 1, например, компьютерную программу, используемую гостиницей для управления бронированием номеров. Демонстрация распознавания образов должна показать учащимся, как: <ul style="list-style-type: none"> определить общие элементы или особенности процессов, или проблем. выявить различия между процессами или проблемами. определить отдельные элементы внутри процессов или проблем. описать шаблоны, которые были определены. делать прогнозы на основе выявленных закономерностей. Демонстрация распознавания образов должна показать учащимся, как: <ul style="list-style-type: none"> определить информацию, необходимую для решения выявленной проблемы. идентифицировать и удалить информацию, которая не нужна для решения выявленной проблемы.

Упражнения	Учебные заметки
	<ul style="list-style-type: none"> ○ описать части проблемы или программы в общих чертах, указав: <ul style="list-style-type: none"> - что нужно вводить. - каковы предполагаемые результаты. - параметры, которые будут меняться. - параметры, которые останутся неизменными. - ключевые действия, которые должна выполнить программа. - повторные процессы, которые программа будет выполнять. ● Используйте метод вопросов и ответов на протяжении всего урока, чтобы вовлечь учащихся в процесс разбиения примера.
Упражнение в малых группах/парах (50 минут)	<ul style="list-style-type: none"> ● Учащиеся практикуют распознавание образов, обобщение и абстракцию образов и расширяют свою работу с Урока 1, используя тот же сценарий, что и в Уроке 1. <ul style="list-style-type: none"> ○ Создайте компьютерную игру. ○ Разбейте проблему. Убедитесь, что вы: <ul style="list-style-type: none"> - определили и описали проблему. - разбили проблему на более мелкие фрагменты. - описали основные особенности проблемы или процесса. ● Учащиеся записывают свои ответы на компьютере или на бумаге.
Обсуждение под руководством преподавателя (20 минут)	<ul style="list-style-type: none"> ● Дайте учащимся возможность оставить отзыв о своей работе остальным участникам группы. ● Позвольте учащимся обсудить, как их работа отличается от работы их одногруппников.
Заключительное упражнение (20 минут)	<ul style="list-style-type: none"> ● Дайте учащимся возможность отработать навык разбиения. Учащиеся должны использовать идеи, которые они разработали в ходе обсуждения, для дополнения или усовершенствования своей работы.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	3 (2 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none"> • понимать необходимость применения навыков решения проблем для анализа проблем. • уметь описывать процессы, необходимые для решения проблемы. • уметь описать, как последовательность, выбор и итерация используются в данной задаче.
-------------------	--

Список ресурсов	<ul style="list-style-type: none"> • Параметры модуля. • Сценарий и работы учащихся с 1 и 2 уроков. • Презентация.
Ключ: СУ : Список упражнений; ШФ : Шаблон формы; СП : слайд презентации	

Упражнения	Учебные заметки
Начальное упражнение (10 минут)	<ul style="list-style-type: none"> Повторим навыки решения проблем, которые были рассмотрены до настоящего времени. Объясните, что на этом уроке учащиеся продолжат изучать навыки решения проблем, в частности: <ul style="list-style-type: none"> последовательность. выбор. итерацию. Используйте примеры из сценария, показанного в демонстрациях на уроках 1 и 2, чтобы проиллюстрировать ваши замечания.
Обсуждение под руководством преподавателя (10 минут)	<ul style="list-style-type: none"> Используйте сессию вопросов и ответов, чтобы проверить понимание учащимися реляционных операторов ($=$, $<$, $>$, \leq, \geq). Объясните, что учащиеся в первую очередь рассмотрят последовательность.
Упражнение в малых группах/парах (30 минут)	<ul style="list-style-type: none"> Используя тот же сценарий, что и в уроках 1 и 2, учащиеся описывают последовательность ключевых задач и процессов. Они должны определить: <ul style="list-style-type: none"> самый эффективный и логичный порядок процессов. порядок операций, необходимых в расчетах. процессы, которые зависят от результатов других процессов или расчетов.
Демонстрация под руководством преподавателя (15 минут)	<ul style="list-style-type: none"> Создайте презентацию, в которой представлена идея выбора (ветвления), If... Then... Else и логические операторы (NOT, AND, OR). Используйте сценарий, чтобы показать, почему используется выбор (ветвление) (), например, различные пункты меню или имеются ли у игрока в игре очки или нет. Примеры того, как ветвление и логическое выражение в компьютерных программах могут быть представлены с помощью этого метода Презентация также должна охватывать концепцию итерации.

Упражнения	Учебные заметки
Упражнение в парах/малых группах (45 минут)	<ul style="list-style-type: none"> Используя тот же сценарий, что и в уроках 1 и 2, учащиеся описывают любые ветвления и итерации, которые потребуются в их программе. Они должны охватывать: <ul style="list-style-type: none"> решения, которые должны быть приняты пользователем или программой. логическую структуру решения и альтернативные результаты. процессы или действия, в которых нужно будет использовать итерацию. тип итерации, которая будет использоваться (например, продолжение запроса ввода данных пользователем, пока ответ не будет правильным).
Заключительное упражнение (10 минут)	<ul style="list-style-type: none"> Закрепите основные пункты обучения, указанные в уроке.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	4–5 (4 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none"> • понимать необходимость применения навыков решения проблем для анализа проблем. • уметь идентифицировать и описать проблему. • уметь описывать процессы, необходимые для решения проблемы. • уметь идентифицировать и описывать результаты, полученные процессом или программой.
-------------------	--

Список ресурсов	<ul style="list-style-type: none"> • Параметры модуля. • Листы бумаги крупного формата • Работа учащихся с прошлого урока. • Пример проблемного сценария для учащихся. • Цветные ручки (для примечаний)
------------------------	--

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (15 минут)	<ul style="list-style-type: none"> Вкратце повторите концепции разбиения, распознавания, обобщения и абстракции шаблонов с помощью сессии вопросов и ответов. Объясните, что на следующих двух уроках учащиеся смогут более детально изучать разработку алгоритмов.
Упражнение в парах/малых группах (2,5 часа)	<ul style="list-style-type: none"> Дайте учащимся новый сценарий для изучения, такой как компьютеризированная проверка в супермаркете или игра типа Space Invade3M. В тех же группах или парах, что и в предыдущих уроках, учащиеся рассматривают «алгоритмы» для разных частей игры (например, для управления космическим кораблем, стрельбы или движений захватчиков). Они записывают алгоритмы на крупноформатных листах бумаги. Объясните, что на ранних стадиях проектирования программного решения «алгоритмы» должны быть только пошаговыми инструкциями, использующими повседневный язык, и их не нужно указывать как полный программный код. Однако учащиеся должны описать решение как можно более подробно.
Упражнение в группах (взаимная проверка и отзывы) (45 минут)	<ul style="list-style-type: none"> На разных этапах в течение двух уроков попросите группы обсудить свою работу с другими небольшими группами. Каждая группа должна дать отзывы о разработанных алгоритмах. Отзывы должны затрагивать полноту и точность алгоритмов каждой группы.
Заключительное упражнение (30 минут)	<ul style="list-style-type: none"> Дайте группам ручки разного цвета на выбор и попросите их прокомментировать свою работу при получении отзывов и любые изменения, которые они вносят в результате полученных отзывов.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	6 (2 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none"> • применять навыки решения проблем для анализа проблем и определения решений, которые могут быть разработаны в компьютерных программах. • описывать вычислительные задачи или процессы, необходимые для решения проблем.
-------------------	---

Список ресурсов	<ul style="list-style-type: none"> • Параметры модуля. • Оценочное упражнение, содержащее краткое описание и сценарий. • Копии содержания для учебной цели А (справочный лист навыков решения проблем).
------------------------	--

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (5 минут)	<ul style="list-style-type: none"> Организуйте учащихся так, чтобы они могли самостоятельно выполнять задания на оценку.
Практическое оценочное упражнение (1 час 55 минут)	<ul style="list-style-type: none"> Дайте учащимся краткое описание сценария, который требует создания компьютерной программы. Сценарий не должен был использоваться ни в одном из предыдущих уроков. Краткое руководство должно иметь достаточный объем, чтобы позволить учащимся продемонстрировать понимание всех пунктов цели обучения А. Учащиеся составляют план решения проблемы. Дайте учащимся копию контента для учебной цели А, чтобы помочь им охватить все вопросы. <p>Примечание: преподаватели должны использовать результаты оценочного упражнения для мониторинга успеваемости учащихся. Это предварительная оценка, которая не учитывается при итоговой оценке модуля.</p>

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	7 (2 часа)

Цели урока	В конце урока учащиеся смогут понимать: <ul style="list-style-type: none">концепции примера процедурного программирования.преимущества и недостатки использования парадигмы процедурного программирования.
-------------------	---

Список ресурсов	<ul style="list-style-type: none">Параметры модуля.Презентация.Компьютеры с доступом в интернет.Учебники.
------------------------	--

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (10 минут)	<ul style="list-style-type: none"> Объясните учащимся, что теперь они будут рассматривать конкретный пример программирования и исследовать процедурное программирование. Объясните, что в рамках итоговой контрольной по данному модулю учащиеся должны будут спланировать, разработать и протестировать программу, использующую данную парадигму.
Презентация преподавателя (10 минут)	<ul style="list-style-type: none"> Покажите презентацию, которая дает обзор функций и характеристик парадигмы процедурного программирования. Познакомьте учащихся со структурой процедурного программирования. Выделите последовательность, выбор и итерацию, где это необходимо.
Самостоятельное упражнение (40 минут)	<ul style="list-style-type: none"> Учащиеся должны провести исследование и сделать заметки об использовании, преимуществах и недостатках процедурного программирования. Если позволяет время, учащиеся должны найти примеры различных процедурных языков программирования (например, Python) и изучить преимущества и недостатки этих языков.
Упражнение в малых группах (40 минут)	<ul style="list-style-type: none"> Реорганизуйте учащихся в небольшие группы, чтобы обсудить полученные ими результаты. Учащиеся обсуждают изученное и сделанные заметки. Во время обсуждения они должны дополнять и расширять свои заметки, в том числе любые важные замечания, которые делают другие учащиеся.
Заключительное упражнение (20 минут)	<ul style="list-style-type: none"> Дайте учащимся возможность оставить отзыв о своей работе остальным участникам группы. Разрешите обсудить, сколько из того, что узнали одни учащиеся, совпадает с тем, что сказали другие.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	8 (2 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none"> • объявить константы и переменные, которые используют соответствующие типы данных. • выбирать и применять математические операторы для простых расчетов. • использовать общие функции для: <ul style="list-style-type: none"> ◦ принятия ввода пользователя. ◦ вывода сообщений на экран. • использовать функции обработки строк, чтобы: <ul style="list-style-type: none"> ◦ преобразовать числовые значения в строки. ◦ преобразовать строки в числовые значения.
-------------------	--

Список ресурсов	<ul style="list-style-type: none"> • Параметры модуля. • Презентация. • Листы с заданиями по программированию. • Компьютеры с интегрированной средой разработки (IDE). • Официальная документация по Python: https://docs.python.org/3/. • Интерактивное руководство по программированию www.learnpython.org/ or www.w3schools.com/python/. • Контрольный опрос в рамках итогового упражнения.
------------------------	---

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (10 минут)	<ul style="list-style-type: none"> Объясните, что на следующем этапе занятий учащиеся познакомятся с основами программирования. Объясните, что занятия будут посвящены процедурной парадигме и, в частности, языку программирования Python.
Презентация преподавателя (20 минут)	<ul style="list-style-type: none"> Покажите презентацию, чтобы представить некоторые ключевые моменты, которые будут рассмотрены в этом уроке, в том числе: <ul style="list-style-type: none"> особенности среды разработки (редактирование кода, запуск кода и т. д.). терминологию (включая константы, переменные, ввод, вывод и строковое целое число). основные используемые команды Python
Самостоятельное упражнение (1 час 10 минут)	<ul style="list-style-type: none"> Дайте учащимся ряд основных задач по программированию, в том числе: <ul style="list-style-type: none"> принятие ввода пользователя. присвоение значения переменной. простые математические уравнения. простую конкатенацию. вывод результатов на экран. преобразование между строкой и целым числом. Примеры команд / функций Python, которые будут использоваться: <ul style="list-style-type: none"> input(). print(). int(). str().

Упражнения	Учебные заметки
Заключительное упражнение (20 минут)	<ul style="list-style-type: none">Проведите краткое обсуждение того, что ученики находят трудным или удивительным в кодировании. Например, тот факт, что все входные данные считаются строками, поэтому введенные числа должны быть преобразованы, прежде чем их можно будет использовать в расчетах.Проведите опрос, чтобы проверить понимание учащимися тем, затронутых в уроке.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	9 (2 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none"> использовать соответствующие структуры данных для их хранения и обработки (справки). выбирать и применять математические операторы (реляционные и булевые операторы). применять выбор при написании кода.
-------------------	---

Список ресурсов	<ul style="list-style-type: none"> Параметры модуля. Примеры кодов. Презентация. Компьютеры с интегрированной средой разработки (IDE). Официальная документация по Python: https://docs.python.org/3/. Интерактивное руководство по программированию www.learnpython.org/ или www.w3schools.com/python/. Листы с заданиями по программированию.
------------------------	--

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (10 минут)	<ul style="list-style-type: none"> Повторите работу, выполненную в предыдущем уроке, и проверьте понимание учащихся с помощью примеров кода.
Обсуждение под руководством преподавателя (15 минут)	<ul style="list-style-type: none"> Изучите концепцию необходимости хранить несколько значений (списков элементов) и необходимость разрешить программам использовать выделение (ветвление). Вернитесь к уроку 3, если в этом есть необходимость.
Демонстрация под руководством преподавателя (10 минут)	<ul style="list-style-type: none"> Проведите презентацию о навыках, которые будут использованы в этом уроке, в том числе: <ul style="list-style-type: none"> как создать список. как вывести содержимое списка. как создавать утверждения IF. о разнице в Python между = and ==.
Самостоятельное упражнение (1 час 10 минут)	<ul style="list-style-type: none"> Дайте учащимся ряд основных задач по программированию, в том числе: <ul style="list-style-type: none"> создание и использование списков. вывод содержимого списка. обеспечение различных процессов / выходов на основе пользовательского ввода. Примеры команд или функций Python для обучения: <ul style="list-style-type: none"> IF. ELSE. ELSEIF (ELIF).
Заключительное упражнение (15 минут)	<ul style="list-style-type: none"> Разбейте учащихся на пары. Учащиеся делятся со своим партнером некоторыми кодами, которые они создали в рамках самостоятельного упражнения. Учащиеся дают своим партнерам отзывы о сильных сторонах их работы и областях для улучшения.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	10 (2 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none"> • применять итерацию при написании кода. • использовать встроенные функции для определения диапазона. • добавлять комментарии к коду для облегчения его понимания.
-------------------	--

Список ресурсов	<ul style="list-style-type: none"> • Параметры модуля. • Компьютеры с интегрированной средой разработки (IDE). • Примеры кодов. • Презентация. • Листы с заданиями по программированию. • Официальная документация по Python: https://docs.python.org/3/. • Интерактивное руководство по программированию www.learnpython.org/ или www.w3schools.com/python/.
------------------------	--

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (10 минут)	<ul style="list-style-type: none"> Повторите работу, выполненную в предыдущем уроке, и проверьте понимание учащихся с помощью примеров кода.
Обсуждение под руководством преподавателя (10 минут)	<ul style="list-style-type: none"> Изучите концепцию необходимости повторения процессов в программе. Используйте сессию вопросов и ответов, чтобы проверить понимание итерации учащимися, как описано в уроке 3.
Демонстрация под руководством преподавателя (20 минут)	<ul style="list-style-type: none"> Покажите презентацию о навыках, которые будут использованы в этом уроке, в том числе: <ul style="list-style-type: none"> как создать цикл «for». как создать цикл «while». как определить диапазон. о разнице в Python между = and ==.
Самостоятельное упражнение (1 часа)	<ul style="list-style-type: none"> Дайте учащимся ряд основных задач по программированию, в том числе: <ul style="list-style-type: none"> создать и использовать разные циклы. решить, какой цикл использовать в конкретной ситуации (например, цикл «while» или цикл «for»). определить числовой диапазон в качестве условия для цикла. объединить выбор (см. урок 9) и итерацию. использовать цикл для вывода содержимого списка (см. урок 9). Примеры команд / функций Python для обучения: <ul style="list-style-type: none"> While. For. range().

Упражнения	Учебные заметки
Итоговое упражнение (20 минут)	<ul style="list-style-type: none">Покажите, как комментировать код (комментарии к коду). Выделите использование #, необходимое для того, чтобы сообщить коду проигнорировать комментарий к этой строке.Объясните важность использования комментариев при написании больших фрагментов кода и объясните, как они помогают другим разработчикам понять логику вашего кода.Учащиеся комментируют код, который они создали во время самостоятельного упражнения, чтобы объяснить, как работает код.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	11 (2 часа)

Цели урока	В конце урока учащиеся смогут: <ul style="list-style-type: none">уметь применять код программирования для решения простых задач.
------------	--

Список ресурсов	<ul style="list-style-type: none">Параметры модуля.Оценочное упражнение.Компьютеры с интегрированной средой разработки (IDE).Официальная документация по Python: https://docs.python.org/3/.Интерактивное руководство по программированию www.learnpython.org/ или www.w3schools.com/python/.
Ключ: СУ : Список упражнений; ШФ : Шаблон формы; СП : слайд презентации	

Упражнения	Учебные заметки
Начальное упражнение (5 минут)	<ul style="list-style-type: none"> Организуйте учащихся так, чтобы они могли самостоятельно выполнять задания на оценку.
Практическое оценочное упражнение (1 час 55 минут)	<ul style="list-style-type: none"> Дайте учащимся ряд основных задач по программированию. Задания должны быть по материалам 8-10 уроков. Задачи должны быть небольшими проблемами кодирования, которые постепенно усложняются на протяжении всей оценки. Задачи не должны требовать создания полноценных рабочих программ. Вместо этого им следует сосредоточиться на тестировании навыков программирования, изученных в уроках 8-10. <p>Примечание: преподаватели должны использовать результаты оценочного упражнения для мониторинга успеваемости учащихся. Это предварительная оценка, которая не учитывается при итоговой оценки модуля.</p>

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	12 (2 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none"> • понимать назначение встроенных функций и библиотеки. • понимать необходимость создавать программы гибкими и непредсказуемыми. • уметь использовать встроенные функции для определения, генерации и использования случайных чисел.
-------------------	---

Список ресурсов	<ul style="list-style-type: none"> • Параметры модуля. • Презентация. • Листы с заданиями и проблемами по программированию. • Компьютеры с интегрированной средой разработки (IDE). • Официальная документация по Python: https://docs.python.org/3/. • Интерактивное руководство по программированию www.learnpython.org/ или www.w3schools.com/python/.
------------------------	---

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (10 минут)	<ul style="list-style-type: none"> Повторите задания, описанные в уроке 10. Обратите особое внимание на понимание учащимися использования функции диапазона. Объясните, что в этом уроке мы рассмотрим некоторые другие встроенные функции, доступные в Python.
Обсуждение под руководством преподавателя (10 минут)	<ul style="list-style-type: none"> Изучите, почему программы следует создавать гибкими и непредсказуемыми. Используйте сессию вопросов и ответов, чтобы побудить учащихся рассмотреть ситуации, когда процессы или действия могут быть случайными.
Демонстрация под руководством преподавателя (20 минут)	<ul style="list-style-type: none"> Покажите презентацию о навыках, которые будут использованы в этом уроке, в том числе: <ul style="list-style-type: none"> концепции числового диапазона. импорт функций и библиотек. использование случайной функции. Узнайте, как компьютер генерирует случайное число, и как использовать диапазоны, чтобы добавить ограничения к генерируемому объему.

Упражнения	Учебные заметки
Самостоятельное упражнение (1 часа)	<ul style="list-style-type: none">• Дайте учащимся ряд основных задач по программированию, в том числе:<ul style="list-style-type: none">◦ создание и использование случайных чисел.◦ определение числового диапазона в качестве параметров для случайного числа.◦ объединение других навыков (например, ввод, вывод, списки и выбор) для создания более сложных программ.• Примеры команд / функций Python, которые будут использоваться:<ul style="list-style-type: none">◦ Import.◦ random().◦ randint().◦ uniform().◦ sample().• Учащиеся должны попытаться решить задачу программирования, используя случайные числа и другие концепции программирования.
Заключительное упражнение (20 минут)	<ul style="list-style-type: none">• В парах учащиеся обсуждают свою работу и дают друг другу отзывы о качестве и эффективности созданного ими кода.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	13 (2 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none"> • понимать необходимость создания качественного кода. • понимать, как использовать локальные и глобальные переменные. • уметь создавать и использовать функции соответствующим образом.
-------------------	---

Список ресурсов	<ul style="list-style-type: none"> • Параметры модуля. • Презентация. • Листы с заданиями по программированию. • Компьютеры с интегрированной средой разработки (IDE). • Официальная документация по Python: https://docs.python.org/3/. • Интерактивное руководство по программированию www.learnpython.org/ или www.w3schools.com/python/.
------------------------	--

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (10 минут)	<ul style="list-style-type: none"> Кратко повторите предыдущий урок и напомните учащимся об использовании встроенных функций. Объясните, что до сих пор учащиеся использовали встроенные функции, но сегодня они научатся создавать свои собственные функции.
Демонстрация под руководством преподавателя (20 минут)	<ul style="list-style-type: none"> Покажите презентацию о навыках, которые будут использованы в этом уроке, в том числе о: <ul style="list-style-type: none"> необходимости создания функций (например, возможность повторного использования, эффективность, снижение риска ошибок). разнице между локальными и глобальными переменными. определении и вызове функций. параметрах и передаче параметров. возвращающих значениях.
Самостоятельное упражнение (1 час 10 минут)	<ul style="list-style-type: none"> Дайте учащимся ряд задач, требующих от них определения, использования и описания функций. Примеры команд / функций Python для обучения: <ul style="list-style-type: none"> def. global.
Заключительное упражнение (20 минут)	<ul style="list-style-type: none"> В парах учащиеся обсуждают свою работу и дают друг другу отзывы о качестве и эффективности созданного ими кода.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	14 (2 часа)

Цели урока	<p>В конце урока учащиеся смогут уметь:</p> <ul style="list-style-type: none"> округлять числа до определенного количества десятичных знаков. усекать числа с помощью встроенных функций. определять местоположение самого большого или самого маленького значения в группе чисел.
-------------------	---

Список ресурсов	<ul style="list-style-type: none"> Параметры модуля. Презентация. Компьютеры с интегрированной средой разработки (IDE). Листы с заданиями по программированию. Официальная документация по Python: https://docs.python.org/3/. Интерактивное руководство по программированию www.learnpython.org/ или www.w3schools.com/python/.
Ключ: СУ : Список упражнений; ШФ : Шаблон формы; СП : слайд презентации	

Упражнения	Учебные заметки
Начальное упражнение (15 минут)	<ul style="list-style-type: none">Начальное задание: вспомните предыдущий урок и объясните другие встроенные функции.Обсудите необходимость указания формата вывода чисел. Используйте сессию вопросов и ответов, чтобы учащиеся могли дать примеры того, когда может понадобиться вывести числа в определенное количество десятичных разрядов или округлить в большую или меньшую сторону.
Демонстрация под руководством преподавателя (20 минут)	<ul style="list-style-type: none">Покажите презентацию о навыках, которые будут использованы в этом уроке, в том числе о:<ul style="list-style-type: none">необходимости округлять или усекать выходы.функции округления и о том, как указать требуемое количество знаков после запятой.импорте «математической» библиотеки.использовании, и различии между, math.trunc (), math.floor () и math.ceil () .использовании максимальной и минимальной функцию.

Упражнения	Учебные заметки
Самостоятельное упражнение (1 час 10 минут)	<ul style="list-style-type: none"> Дайте учащимся ряд заданий по программированию, чтобы охватить навыки, пройденные на этом уроке, включая: <ul style="list-style-type: none"> округление и усечение положительных и отрицательных чисел. выбор и использование соответствующих функций усечения и округления для решения заданных проблем. вывод максимального или минимального числа в списке. Примеры команд / функций Python, которые будут использоваться: <ul style="list-style-type: none"> import math. round(). math.trunc(). math.floor(). math.ceil(). max(). min().
Заключительное упражнение (15 минут)	<ul style="list-style-type: none"> Используйте сессию вопросов и ответов, чтобы проверить понимание учащимися темы итерации, как описано в уроке.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	15 (2 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none"> • понимать необходимость проверки формата или содержимого строки. • в состоянии проверить, написана ли строка в верхнем или нижнем регистре. • в состоянии проверить, содержит ли строка числовые значения. • в состоянии преобразовать строку из верхнего регистра в нижний регистр. • в состоянии преобразовать строку из нижнего регистра в верхний регистр.
-------------------	---

Список ресурсов	<ul style="list-style-type: none"> • Параметры модуля. • Презентация. • Компьютеры с интегрированной средой разработки (IDE). • Листы с заданиями по программированию. • Контрольный опрос в рамках итогового упражнения. • Официальная документация по Python: https://docs.python.org/3/. • Интерактивное руководство по программированию www.learnpython.org/ or www.w3schools.com/python/.
------------------------	---

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (10 минут)	<ul style="list-style-type: none"> Повторите урок 8 о преобразовании строки в число. Проверьте понимание материала учеником из предыдущего урока. Объясните, что на сегодняшнем уроке мы продолжим изучение дополнительных встроенных функций.
Обсуждение под руководством преподавателя (10 минут)	<ul style="list-style-type: none"> Обсуждение под руководством преподавателя: изучите необходимость сделать код устойчивым, включая проверку пользовательского ввода. Используйте сессию вопросов и ответов, чтобы учащиеся придумали ситуации, когда им может понадобиться проверить ввод пользователя. В частности, предложите им подумать, когда им может понадобиться проверить регистр букв или проверить, был ли включен номер.
Демонстрация под руководством преподавателя (20 минут)	<ul style="list-style-type: none"> Покажите презентацию о навыках, которые будут использованы в этом уроке, в том числе: <ul style="list-style-type: none"> использование встроенных функций Python для проверки текста или числа. преобразование между строчными и прописными буквами.
Самостоятельное упражнение (1 час 10 минут)	<ul style="list-style-type: none"> Дайте учащимся ряд заданий по программированию, чтобы охватить навыки, пройденные на этом уроке, включая: <ul style="list-style-type: none"> изъятие пользовательского ввода и сохранение в переменную. проверку наличия числовых значений в строке. преобразование сохраненной строки из верхнего регистра в нижний регистр. преобразование сохраненной строки из нижнего регистра в верхний регистр. выбор для проверки ввода и предоставления обратной связи пользователю.

Упражнения	Учебные заметки
	<ul style="list-style-type: none">• Примеры команд / функций Python, которые будут использоваться:<ul style="list-style-type: none">◦ isupper().◦ islower().◦ upper().◦ lower().◦ isalpha().
Заключительное упражнение (10 минут)	<ul style="list-style-type: none">• Все учащиеся проходят контрольный опрос по темам, которые они прошли на уроке.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	16 (2 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none"> • понимать необходимость использования нескольких элементов данных в программе. • уметь использовать список / массив в программе. • в состоянии использовать словарь / карту в программе.
-------------------	---

Список ресурсов	<ul style="list-style-type: none"> • Параметры модуля. • Презентация. • Компьютеры с интегрированной средой разработки (IDE). • Листы с заданиями по программированию. • Официальная документация по Python: https://docs.python.org/3/. • Интерактивное руководство по программированию www.learnpython.org/ или www.w3schools.com/python/.
Ключ: СУ : Список упражнений; ШФ : Шаблон формы; СП : слайд презентации	

Упражнения	Учебные заметки
Начальное упражнение (10 минут)	<ul style="list-style-type: none">Используйте сессию вопросов и ответов, чтобы вспомнить основное использование списков / массивов и проверить понимание учащимися этой темы (при необходимости обращайтесь к уроку 9).Объясните, что на этом уроке учащиеся продолжат изучать использование списков и связанных структур данных.
Демонстрация под руководством преподавателя (20 минут)	<ul style="list-style-type: none">Покажите презентацию о том,<ul style="list-style-type: none">как индексация используется для ссылки на местоположение элемента в списке / массиве.как настроить список / массив.как добавить данные в список / массив во время выполнения, используя результаты расчетов и пользовательского ввода.как добавить данные в список / массив.как вставить данные в определенном месте.как создавать «всплывающие» данные.

Упражнения	Учебные заметки
Самостоятельное упражнение (1 час 10 минут)	<ul style="list-style-type: none"> ● Дайте учащимся ряд заданий по программированию, чтобы охватить навыки, пройденные на этом уроке, включая: <ul style="list-style-type: none"> ○ изъятие пользовательского ввода и добавление его в список / массив. ○ обработку данных в программе с использованием списка / массива. ○ добавление данных в список. ○ вставку данных в определенных местах. ○ удаление данных. ○ «всплывающие» данные. ● Примеры команд / функций Python, которые будут использоваться: <ul style="list-style-type: none"> ○ index(). ○ append(). ○ insert(). ○ remove(). ○ count(). ○ pop(). ○ sort().
Заключительное упражнение (10 минут)	<ul style="list-style-type: none"> ● Учащиеся работают в парах, обмениваясь друг с другом отзывами о качестве и эффективности кода.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	17 (2 часа)

Цели урока	В конце урока учащиеся смогут: <ul style="list-style-type: none"> • понимать необходимость использования нескольких элементов данных в программе. • уметь использовать список / массив в программе. • уметь использовать словарь / карту в программе. • уметь использовать функции для обработки данных в списке / массиве.
-------------------	---

Список ресурсов	<ul style="list-style-type: none"> • Параметры модуля. • Презентации. • Компьютеры с интегрированной средой разработки (IDE). • Листы с заданиями по программированию. • Официальная документация по Python: https://docs.python.org/3/. • Интерактивное руководство по программированию www.learnpython.org/ или www.w3schools.com/python/.
------------------------	--

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (10 минут)	<ul style="list-style-type: none"> Краткое повторение предыдущего урока. Используя сессию вопросов и ответов, подытожьте и проверьте понимание материала учащимися (при необходимости обращайтесь к материалам урока 16). Объясните, что на этом уроке учащиеся смогут больше изучать использование списков и связанных структур данных.
Демонстрация под руководством преподавателя (15 минут)	<ul style="list-style-type: none"> Покажите презентацию о навыках, которые будут использованы в этом уроке, в том числе: <ul style="list-style-type: none"> повторение использования индексации для ссылки на местоположение элемента в списке / массиве. проверка наличия в списке / массиве конкретного символа, слова или значения. возвращение местоположения определенного символа, слова или значения. сортировка элементов в списке / массиве в порядке возрастания или убывания.
Самостоятельное упражнение (35 минут)	<ul style="list-style-type: none"> Дайте учащимся ряд заданий по программированию, чтобы охватить навыки, пройденные на этом уроке. Они должны основываться на тех, которые использовались в предыдущем уроке, и вводить такие дополнительные уровни сложности, как: <ul style="list-style-type: none"> изъятие пользовательского ввода и добавление его в список / массив. проверка наличия в списке / массиве конкретного символа, слова или значения. возвращение местоположения определенного символа, слова или значения. сортировка элементов в списке / массиве в порядке возрастания или убывания.

Упражнения	Учебные заметки
	<ul style="list-style-type: none">Примеры команд / функций Python, которые будут использоваться:<ul style="list-style-type: none">index().append().sort().for loops.in.not in.count().
Демонстрация под руководством преподавателя (10 минут)	<ul style="list-style-type: none">Покажите презентацию о различиях между словарем / картой и списком / массивом в Python и о том, как использовать их в коде.
Самостоятельное упражнение (40 минут)	<ul style="list-style-type: none">Дайте учащимся ряд заданий по программированию, чтобы охватить навыки, пройденные на этом уроке.
Заключительное упражнение (10 минут)	<ul style="list-style-type: none">Организуйте учеников в пары. Учащиеся работают вместе, чтобы обсудить свой код и дать друг другу отзывы о его качестве и эффективности.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	18 (2 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none"> понять необходимость манипулировать предметами внутри строки. уметь использовать функции для манипулирования данными, хранящимися в виде строк.
-------------------	--

Список ресурсов	<ul style="list-style-type: none"> Параметры модуля. Презентация. Компьютеры с интегрированной средой разработки (IDE). Листы с заданиями по программированию. Официальная документация по Python: https://docs.python.org/3/. Интерактивное руководство по программированию www.learnpython.org/ или www.w3schools.com/python/.
------------------------	--

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (10 минут)	<ul style="list-style-type: none"> Посмотрите, как ученики понимают вопросы индексации (при необходимости обращаясь к материалам, использованным в уроках 16 и 17). Объясните, что на этом уроке учащиеся увидят, как навыки, которые они приобрели при использовании индексирования, также можно использовать для манипулирования строками.
Демонстрация под руководством преподавателя (15 минут)	<ul style="list-style-type: none"> Покажите презентацию о навыках, которые будут использованы в этом уроке, в том числе: <ul style="list-style-type: none"> как расположить каждый символ в строке с использованием индекса, а также элемент в списке. как разделить строку на отдельные символы. как проверить, находится ли конкретная буква или слово в строке. как вывести местоположение конкретного символа в строке. как проверить длину строки.
Самостоятельное упражнение (1 час 10 минут)	<ul style="list-style-type: none"> Дайте учащимся ряд заданий по программированию, чтобы охватить навыки, пройденные на этом уроке, включая: <ul style="list-style-type: none"> разделение строки на отдельные символы. расположение символов в определенных точках в строке и проверка того, находится ли конкретная буква или слово в строке. проверка соответствия длины строки заранее определенным требованиям и выдача соответствующих сообщений пользователям. Примеры команд / функций Python, которые будут использоваться: <ul style="list-style-type: none"> split(). len(). for loops. in. not in. count()

Упражнения	Учебные заметки
Заключительное упражнение (25 минут)	<ul style="list-style-type: none">• Работая в парах, учащиеся меняют один из своих фрагментов кода с текущего урока на один из фрагментов кода своего партнера. Учащиеся пытаются испортить программу своего партнера.• Учащиеся делают заметки о результатах проделанной работы. Объясните, что они будут изучать проблему сбоя программы на следующем уроке.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	19 (2 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none">• понимать необходимость создания надежного программного кода.• уметь применять методы проверки к программному коду.
------------	---

Список ресурсов	<ul style="list-style-type: none">• Параметры модуля.• Презентация.• Компьютеры с интегрированной средой разработки (IDE).• Листы с заданиями по программированию.• Пример требований к коду.• Справочные листы кода (включающие циклы, обработку строк, индексацию).• Официальная документация по Python: https://docs.python.org/3/.• Интерактивное руководство по программированию www.learnpython.org/ или www.w3schools.com/python/.
-----------------	--

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы ; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (10 минут)	<ul style="list-style-type: none"> Подытожите итоговое задание из урока 18. Попросите учащихся описать примеры, которые привели к сбою кода. Они могут продемонстрировать свои коды.
Обсуждение под руководством преподавателя (10 минут)	<ul style="list-style-type: none"> Изучить важность надежных программ. Обсудите конкретные сценарии, в которых программы должны будут проверять пользовательский ввод. Попросите учеников изучить проблемы, которые могут возникнуть в случае сбоя важной программы.
Демонстрация под руководством преподавателя (15 минут)	<ul style="list-style-type: none"> Покажите презентацию о навыках, которые будут использованы в этом уроке, в том числе: <ul style="list-style-type: none"> краткое повторение использования выбора и итерации. ограничение количества принятых вариантов. принятие только тех значений, которые находятся в заданном диапазоне. принятие только тех входных данных, которые соответствуют заданным критериям длины. принятие только данных определенного типа (например, только текст, только номер). принятие только тех данных, которые соответствуют определенным требованиям к формату (например, должны содержать символ @).
Самостоятельное упражнение (30 минут)	<ul style="list-style-type: none"> Дайте учащимся ряд заданий по программированию, чтобы охватить навыки, пройденные на этом уроке, включая: <ul style="list-style-type: none"> использование встроенные или пользовательские функции для применения методов проверки к программному коду. итерацию, чтобы программа не продолжалась до тех пор, пока не будет получен приемлемый ввод.

Упражнения	Учебные заметки
	<ul style="list-style-type: none">Примеры команд / функций Python, которые будут использоваться:<ul style="list-style-type: none">циклы «for».циклы «while».If...Else.In.not in.len().
Самостоятельное упражнение (40 минут)	<ul style="list-style-type: none">Дайте учащимся требования к разделу кода (включая требования к валидации). Например:<ul style="list-style-type: none">Создайте программный код для простой системы заказа пиццы. Система должна позволять пользователям выбирать маленькую, среднюю или большую пиццу и необходимое количество пиццы. Программа должна рассчитать стоимость заказа и не должна непредвиденно завершать работу, если введена неправильная опция.
Заключительное упражнение (15 минут)	<ul style="list-style-type: none">Работая в парах, учащиеся меняют один из своих фрагментов кода с текущего урока на один из фрагментов кода своего партнера. Ученики пытаются вызвать сбой программы и предоставляют обратную связь.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	20 (2 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none"> • понимать необходимость использования файлов, которые являются внешними по отношению к программе, для хранения данных. • уметь открывать, читать и записывать во внешние файлы.
-------------------	---

Список ресурсов	<ul style="list-style-type: none"> • Параметры модуля. • Презентация. • Компьютеры с интегрированной средой разработки (IDE). • Листы с заданиями по программированию. • Контрольный опрос в рамках итогового упражнения. • Официальная документация по Python: https://docs.python.org/3/. • Интерактивное руководство по программированию www.learnpython.org/ или www.w3schools.com/python/.
------------------------	--

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (15 минут)	<ul style="list-style-type: none"> Повторите использование структур данных (массив / список, словарь). Объясните, что в некоторых случаях, например, когда у вас большой объем данных, более эффективно хранить данные вне программного кода.
Демонстрация под руководством преподавателя (20 минут)	<ul style="list-style-type: none"> Покажите презентацию о навыках, которые будут использованы в этом уроке, в том числе: <ul style="list-style-type: none"> открытие внешнего файла в разных режимах (например, чтение, редактирование). создание внешнего файла для использования программой. чтение из внешнего файла и запись в него.
Самостоятельное упражнение (1 час 10 минут)	<ul style="list-style-type: none"> Дайте учащимся ряд заданий по программированию, чтобы охватить навыки, пройденные на этом уроке, включая: <ul style="list-style-type: none"> использование встроенных или пользовательских функций для применения методов проверки к программному коду. итерацию, чтобы программа не продолжалась до тех пор, пока не будет получен приемлемый ввод. Примеры команд / функций Python, которые будут использоваться: <ul style="list-style-type: none"> open(). write(). close(). read(). readline(). line.split().
Заключительное упражнение (15 минут)	<ul style="list-style-type: none"> Дайте учащимся возможность пройти тест, охватывающий темы, которые они рассмотрели в этом уроке.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	21 (2 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none"> • применять навыки решения проблем для анализа проблем и определения решений, которые могут быть разработаны в компьютерные программы. • описывать вычислительные задачи или процессы, необходимые для решения проблем.
-------------------	--

Список ресурсов	<ul style="list-style-type: none"> • Параметры модуля. • Оценочное упражнение • Компьютеры с интегрированной средой разработки (IDE). • Справочный лист по кодированию Python или справочник по кодам и синтаксису.
Ключ: СУ : Список упражнений; ШФ : Шаблон формы; СП : слайд презентации	

Упражнения	Учебные заметки
Начальное упражнение (5 минут)	<ul style="list-style-type: none">Организуйте учащихся так, чтобы они могли самостоятельно выполнять задания на оценку.
Практическое оценочное упражнение (1 час 55 минут)	<ul style="list-style-type: none">Дайте учащимся ряд основных задач по программированию.Они должны создать код для решения каждой из указанных проблем. Проблемы должны быть небольшими, а не такими, что требуют создания больших сложных программ.Предоставьте достаточное количество задач, чтобы учащиеся могли использовать широкий спектр навыков, которые они изучили в уроках 8–19.Примеры проблемных тем включают:<ul style="list-style-type: none">два шестигранных симулятора.конвертер валют.проверка имени пользователя и электронной почты.оценка теста для оценки конвертора. <p>Примечание: преподаватели должны использовать оценочное упражнение для мониторинга успеваемости учащихся. Это предварительная оценка, которая не учитывается при итоговой оценке модуля.</p>

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	22 (2 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none"> • понимать необходимость тестирования кода во время разработки. • понимать важность итеративной разработки при написании кода для создания решений. • писать план тестирования, чтобы проверить программу на соответствие требованиям.
-------------------	--

Список ресурсов	<ul style="list-style-type: none"> • Параметры модуля. • Презентация. • Компьютеры с интегрированной средой разработки (IDE). • Частично заполненный программный код и программные требования. • Шаблон плана испытаний. • Официальная документация по Python: https://docs.python.org/3/. • Интерактивное руководство по программированию www.learnpython.org/ или www.w3schools.com/python/.
------------------------	---

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (10 минут)	<ul style="list-style-type: none"> Перечислите ключевые моменты из уроков 18 и 19, касающиеся необходимости обеспечения надежности программ.
Обсуждение под руководством преподавателя (10 минут)	<ul style="list-style-type: none"> Объясните: валидация, описанная в предыдущих уроках, является лишь одной из составляющих надежных программ. Исследуйте важность надежных программ и роль тестирования. Используйте сессию вопросов и ответов, чтобы подчеркнуть тот факт, что они, вероятно, уже провели некоторое «тестирование» при создании своего кода на предыдущих уроках, например, когда они написали новый фрагмент кода и вводили данные, чтобы увидеть, работает ли он. Это является тестированием!
Демонстрация под руководством преподавателя (20 минут)	<ul style="list-style-type: none"> Покажите презентацию о навыках, которые будут использованы в этом уроке, в том числе о: <ul style="list-style-type: none"> важности формального планирования и записи процесса тестирования. цели плана тестирования и как он используется. различных типах тестовых данных, включая действительные, допустимые экстремальные, недействительные, недопустимые экстремальные и ошибочные. важности проверки функциональности, надежности и точности результатов / расчетов.
Самостоятельное упражнение (1 часа)	<ul style="list-style-type: none"> Дайте учащимся спецификацию требований программы и часть кода. Код должен соответствовать некоторым требованиям, но не всем. Программный код при запуске должен быть функциональным (т.е. работать), однако он может быть не полностью устойчивым (например, не все входные данные будут проверены, поэтому он может привести к неправильным выходным данным или аварийному завершению при вводе некоторых непредвиденных данных). Дайте учащимся шаблон плана тестирования. Учащиеся создают подробный план тестирования для кода, чтобы проверить, насколько он соответствует данной спецификации.

Упражнения	Учебные заметки
Заключительное упражнение (20 минут)	<ul style="list-style-type: none">• Заключительное упражнение: в парах учащиеся обсуждают свои планы тестирования. Они должны быть готовы объяснить или обосновать свой выбор тестов и данных испытаний.• Дайте учащимся время добавить дополнительные тесты в свой план тестирования, основываясь на обсуждении с партнером.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	23 (2 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none">• понимать необходимость тестирования кода во время разработки.• понимать важность итеративной разработки при написании кода для создания решений.• систематически применять план тестирования и связанные с ним тестовые данные.
------------	---

Список ресурсов	<ul style="list-style-type: none">• Параметры модуля.• Презентация.• Компьютеры с интегрированной средой разработки (IDE).• Частично заполненный программный код и программные требования из предыдущего урока.• Планы тестирования учащихся из предыдущего урока.• Официальная документация по Python: https://docs.python.org/3/.• Интерактивное руководство по программированию www.learnpython.org/ или www.w3schools.com/python/.
-----------------	--

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (10 минут)	<ul style="list-style-type: none"> Повторите ключевые моменты из предыдущего урока. Используйте сессию вопросов и ответов, чтобы повторить с учащимися различные типы тестовых данных и напомнить о важности тестирования.
Демонстрация под руководством преподавателя (20 минут)	<ul style="list-style-type: none"> Покажите презентацию о навыках, которые будут использованы в этом уроке, в том числе: <ul style="list-style-type: none"> тестирование кода и запись результатов. выявление ошибок или неожиданного поведения в программе. исправление или улучшение кода. улучшение записи и повторное тестирование кода.
Самостоятельное упражнение (1 час 10 минут)	<ul style="list-style-type: none"> Дайте учащимся спецификацию требований программы и фрагмент кода, который они использовали в предыдущем уроке. Учащиеся применяют свой план тестирования к данному коду, чтобы выявить ошибки и улучшить код по мере необходимости.
Заключительное упражнение (20 минут)	<ul style="list-style-type: none"> В заключение учащиеся комментируют код, чтобы определить, как он работает и какие изменения они внесли.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	24 (2 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none"> • применять навыки решения проблем для анализа проблем и определения решений, которые могут быть разработаны в компьютерных программах. • описывать вычислительные задачи или процессы, необходимые для решения проблем.
-------------------	---

Список ресурсов	<ul style="list-style-type: none"> • Параметры модуля. • Презентация. • Компьютеры с интегрированной средой разработки (IDE). • Руководство по решению задач. • Копии содержания для учебной цели А. • Справочный лист навыков решения проблем.
Ключ: СУ : Список упражнений; ШФ : Шаблон формы; СП : слайд презентации	

Упражнения	Учебные заметки
Начальное упражнение (10 минут)	<ul style="list-style-type: none"> Объясните учащимся, что в течение следующих четырех уроков они выполнят задание, которое позволит им попрактиковать изученный в данном модуле материал.
Обсуждение под руководством преподавателя (20 минут)	<ul style="list-style-type: none"> Покажите презентацию и используйте сессию вопросов и ответов, чтобы напомнить учащимся о навыках, которые они развили в учебной цели А (Уроки 1–6), включая: <ul style="list-style-type: none"> разбиение. распознавание шаблона. обобщение и абстракция шаблона. разработка алгоритмов, включая методы, используемые для этого.
Самостоятельное упражнение (1 час 10 минут)	<ul style="list-style-type: none"> Дайте учащимся компьютерную проблему среднего размера. Краткое руководство должно иметь достаточный объем, чтобы позволить ученикам продемонстрировать понимание всех пунктов в цели обучения А. Учащиеся составляют план решения проблемы. Дайте учащимся копию содержания для учебной цели А, чтобы помочь им охватить все вопросы.
Заключительное упражнение (20 минут)	<ul style="list-style-type: none"> Заключительное упражнение: парами учащиеся обсуждают свои планы. Они должны быть готовы объяснить или обосновать свой выбор и предоставить обратную связь. Если позволяет время, учащиеся предоставляют друг другу отзывы и используют их для внесения изменений в свой план.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	25 (2 часа)

Цели урока	В конце урока учащиеся смогут: <ul style="list-style-type: none">уметь выбирать соответствующие тесты и тестовые данные для проверки функциональности решения.
------------	--

Список ресурсов	<ul style="list-style-type: none">Параметры модуля.Работа учащихся с прошлого урока.Презентация.Руководство по решению задач.Шаблон плана испытаний.Компьютеры с интегрированной средой разработки (IDE).
-----------------	--

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (15 минут)	<ul style="list-style-type: none"> Повторите предыдущий урок и цели данной серии уроков. Покажите учащимся заметки, которые они сделали о своих планах в конце предыдущего урока. Дайте учащимся некоторое время, чтобы внести какие-либо дополнительные изменения на основе их отзывов от одногруппников.
Обсуждение под руководством преподавателя (15 минут)	<ul style="list-style-type: none"> Покажите презентацию и используйте сессию вопросов и ответов, чтобы напомнить учащимся о навыках, которые они развили в данном уроке, включая: <ul style="list-style-type: none"> важность итеративной разработки. необходимость проводить тестирование на протяжении всей разработки программы. причины, по которым часть тестирования следует планировать до начала разработки. необходимость строить и адаптировать предварительный план тестирования и данные тестирования по мере разработки программы.
Самостоятельное упражнение (30 минут)	<ul style="list-style-type: none"> Упражнение в парах/группах: парами учащиеся обсуждают свои планы. Предложите учащимся использовать план, который они создали на последнем уроке, вместе с кратким описанием проблемы, с тем, чтобы направить учащихся по требованиям в плане тестирования.
Упражнение в парах/малых группах (20 минут)	<ul style="list-style-type: none"> Учащиеся обсуждают свои планы тестирования в парах. Они должны быть готовы объяснить или обосновать свой выбор и предоставить обратную связь.

Упражнения	Учебные заметки
Самостоятельное упражнение (40 минут)	<ul style="list-style-type: none">• Учащиеся используют свой план из предыдущего урока, чтобы начать создавать программу.• Убедитесь, что учащиеся тестируют свой код по мере его разработки.• Учащиеся должны стремиться использовать тесты, которые они разработали в своем плане тестирования. Кроме того, любое тестирование, которое они проводят сейчас и которое не входит в их план, должно быть добавлено в план по мере его выполнения.• Напомните учащимся записывать результаты тестирования и любые изменения, которые они вносят при тестировании.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	26 -27 (4 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none"> использовать стандартные структуры программирования и принятые соглашения о программировании для создания кода для решения выявленных проблем. применять понимание итеративной разработки при написании кода для выработки решений.
-------------------	--

Список ресурсов	<ul style="list-style-type: none"> Параметры модуля. Работа учащихся с прошлого урока. Руководство по решению задач. Шаблон плана испытаний. Справочник по языку и синтаксису. Компьютеры с интегрированной средой разработки (IDE). Официальная документация по Python: https://docs.python.org/3/. Интерактивное руководство по программированию www.learnpython.org/ или www.w3schools.com/python/.
------------------------	--

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (10 минут)	<ul style="list-style-type: none"> Повторите предыдущий урок и цели данной серии уроков. Объясните учащимся, что на этом и следующем уроках они продолжат разрабатывать решение данной проблемы. Убедитесь, что учащиеся тестируют свой код по мере его разработки.
Оценочное упражнение (3 час 50 минут)	<ul style="list-style-type: none"> Учащиеся продолжают разрабатывать свою программу для решения данной проблемы, используя при необходимости справочные руководства. Поощряйте учащихся просить отзывы других членов группы на разных этапах разработки. Учащиеся должны следить за тем, чтобы они записывали выполненные ими тесты, результаты тестов и любые изменения, внесенные в их код в результате исходов тестов. Используйте результаты уроков 24–27, чтобы определить сильные стороны учащихся и области для улучшения. В конце данных двух уроков учащиеся представляют свои работы (в любом подходящем формате). Представленные материалы должны включать: <ul style="list-style-type: none"> план программы. план тестирования / журнал испытаний. файл программы. программный код. Оцените работы учащихся и дайте им подробные отзывы о сильных и слабых сторонах их работы. <p>Примечание: это не является окончательной оценкой и не должно использоваться в качестве доказательства их окончательной оценки. Преподаватели должны использовать результаты оценочного упражнения для мониторинга успеваемости учащихся. Это предварительная оценка, которая не учитывается при итоговой оценке модуля.</p>

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	28–29 (4 часа)

Цели урока	В конце урока учащиеся смогут:
	<ul style="list-style-type: none"> определить сильные и слабые стороны в предметных знаниях и навыках. продемонстрировать независимость и саморазвитие.

Список ресурсов	<ul style="list-style-type: none"> Параметры модуля. Отзывы преподавателя о работе по урокам 24–27. Работа учащихся по урокам 24–27. Практические упражнения по программированию. Компьютеры с интегрированной средой разработки (IDE). Официальная документация по Python: https://docs.python.org/3/. Интерактивное руководство по программированию www.learnpython.org/ или www.w3schools.com/python/.
------------------------	--

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (10 минут)	<ul style="list-style-type: none">Предоставьте учащимся копии отзывов преподавателя о работе, проведённой в уроках 24–32.
Самостоятельное упражнение (30 минут)	<ul style="list-style-type: none">Учащиеся читают отзывы преподавателя. Затем они определяют области, в которых им необходимо совершенствоваться, а также тематические области, в которых им требуется дальнейшая помощь, и навыки, которые, по их мнению, им необходимы для дальнейшей практики.
Самостоятельная работа/работа в парах (3 час 20 минут)	<ul style="list-style-type: none">Учащиеся работают над различными заданиями или учебными пособиями, чтобы попрактиковаться и улучшить слабые стороны.Позвольте учащимся самостоятельно обдумывать действия, которые они совершают. Задания могут включать в себя:<ul style="list-style-type: none">использование онлайн программирования.дальнейшее развитие оцененной работы.обсуждение в парах и поддержку.пересмотр задач программирования из предыдущих уроков.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	30 -32 (6 часа)

Цели урока	В конце урока учащиеся смогут: <ul style="list-style-type: none"> • планировать решения выявленной проблемы. • создавать программный код для решения выявленной проблемы. • создавать программный код для решения выявленной проблемы.
-------------------	---

Список ресурсов	<ul style="list-style-type: none"> • Параметры модуля. • Руководство по решению задач. • Шаблон плана испытаний. • Справочник по языку и синтаксису. • Компьютеры с интегрированной средой разработки (IDE). • Официальная документация по Python: https://docs.python.org/3/. • Интерактивное руководство по программированию www.learnpython.org/ или www.w3schools.com/python/.
Ключ: СУ : Список упражнений; ШФ : Шаблон формы; СП : слайд презентации	

Упражнения	Учебные заметки
Начальное упражнение (15 минут)	<ul style="list-style-type: none">Начальное упражнение: вспомните процесс, которому ученики следовали в уроках 24–27.Организуйте учащихся так, чтобы они могли самостоятельно выполнять задания на оценку.Дайте учащимся немного времени для прочтения краткого изложения программы и проясните любые вопросы.
Практическое оценочное упражнение (5 час 45 минут)	<ul style="list-style-type: none">Учащиеся планируют, разрабатывают и тестируют компьютерную программу для решения подробно изложенной в данном задании проблемы.Задача должна быть достаточно объёмной для того, чтобы учащиеся смогли попрактиковаться и продемонстрировать ряд навыков программирования, перечисленных в спецификации.В этом упражнении вы, возможно, захотите дать краткое изложение проблемы, посвященное конкретным областям, в которых учащиеся должны развивать свои навыки. Это даст учащимся возможность совершенствоваться до итоговой контрольной. <p>Примечание: преподаватели должны использовать результаты оценочного упражнения для мониторинга успеваемости учащихся. Это предварительная оценка, которая не учитывается при итоговой оценке модуля.</p>

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	33–34 (4 часа)

Цели урока	В конце урока учащиеся смогут: <ul style="list-style-type: none"> определить сильные и слабые стороны в предметных знаниях и навыках. продемонстрировать независимость и саморазвитие.
-------------------	--

Список ресурсов	<ul style="list-style-type: none"> Параметры модуля. Отзывы преподавателя о работе по урокам 30-32. Практические упражнения по программированию. Компьютеры с интегрированной средой разработки (IDE). Официальная документация по Python: https://docs.python.org/3/. Интерактивное руководство по программированию www.learnpython.org/ или www.w3schools.com/python/.
------------------------	--

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (10 минут)	<ul style="list-style-type: none">Предоставьте учащимся копии отзывов преподавателя о работе, проведённой в уроках 30–32.
Самостоятельное упражнение (30 минут)	<ul style="list-style-type: none">Учащиеся читают отзывы преподавателя. Затем они определяют области, в которых им необходимо совершенствоваться, а также тематические области, в которых им требуется дальнейшая помощь, и навыки, которые, по их мнению, им необходимы для дальнейшей практики.
Самостоятельное упражнение (3 час 20 минут)	<ul style="list-style-type: none">Учащиеся работают над различными заданиями или учебными пособиями, чтобы попрактиковаться и усилить слабые стороны.Позвольте учащимся самостоятельно обдумывать действия, которые они совершают. Задания могут включать в себя:<ul style="list-style-type: none">использование онлайн программирования.дальнейшее развитие оцененной работы.обсуждение в парах и поддержку.пересмотр задач программирования из предыдущих уроков.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	35–39 (10 часов)

Цели урока	В конце урока учащиеся смогут:
	<ul style="list-style-type: none"> • выполнить практическую самостоятельную работу.

Список ресурсов	<ul style="list-style-type: none"> • Практическое оценочное упражнение. • Шаблон плана испытаний. • Справочник по языку и синтаксису. • Компьютеры с интегрированной средой разработки (IDE).
------------------------	---

Ключ: **СУ**: Список упражнений; **ШФ**: Шаблон формы; **СП**: слайд презентации

Упражнения	Учебные заметки
Начальное упражнение (15 минут)	<ul style="list-style-type: none">• Начальное упражнение: объясните, что в следующих пяти уроках учащиеся выполнят практическую самостоятельную работу для подготовки к итоговой контрольной.• Объясните, что все работы для этой самостоятельной работы должны быть их собственными и что работать в парах не разрешается.
Практическое оценочное упражнение (9 час 45 минут)	<ul style="list-style-type: none">• Учащиеся практикуют самостоятельную работу, похожий на итоговую контрольную. <p>Примечание: используемое задание на оценку должно соответствовать структуре, аналогичной итоговой оценке, и должно оцениваться по тем же критериям. Однако используемое оценочное упражнение не должно быть из числа встречающихся в Рабочей тетради для оценки.</p>

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	40 (2 часа)

Цели урока	<p>В конце урока учащиеся смогут:</p> <ul style="list-style-type: none"> определить сильные и слабые стороны в предметных знаниях и навыках. продемонстрировать независимость и саморазвитие.
-------------------	---

Список ресурсов	<ul style="list-style-type: none"> Параметры модуля. Работа учащихся по урокам 35-39. Отзывы преподавателя о практических самостоятельных работах из уроков 35-39. Практические упражнения по программированию. Компьютеры с интегрированной средой разработки (IDE). Официальная документация по Python: https://docs.python.org/3/. Интерактивное руководство по программированию www.learnpython.org/ или www.w3schools.com/python/.
Ключ: СУ : Список упражнений; ШФ : Шаблон формы; СП : слайд презентации	

Упражнения	Учебные заметки
Начальное упражнение (10 минут)	<ul style="list-style-type: none">• Дайте учащимся проверенные копии их практической самостоятельной работы.
Самостоятельное упражнение (30 минут)	<ul style="list-style-type: none">• Учащиеся читают отзывы преподавателя. Затем они определяют области, в которых им необходимо совершенствоваться, а также тематические области, в которых им требуется дальнейшая помощь, и навыки, которые, по их мнению, им необходимы для дальнейшей практики.
Самостоятельное упражнение (1 час 20 минут)	<ul style="list-style-type: none">• Учащиеся работают над различными заданиями или учебными пособиями, чтобы попрактиковаться и усилить слабые стороны.• Позвольте учащимся самостоятельно обдумывать действия, которые они совершают. Задания могут включать в себя:<ul style="list-style-type: none">◦ использование онлайн программирования.◦ дальнейшее развитие оцененной работы,◦ обсуждение в парах и поддержку.◦ пересмотр задач программирования из предыдущих уроков.

План урока

Квалификация	Pearson ВТЕС Узбекистан Уровень 4 Квалификации в сфере разработки программного обеспечения
Модуль	Модуль 1 Введение в программирование
Номер урока	41–45 (10 часов)

Цели урока	Оценивание целей обучения А, В и С.
-------------------	-------------------------------------

Список ресурсов	<ul style="list-style-type: none"> Параметры модуля. Рабочая тетрадь для оценки. Компьютеры с интегрированной средой разработки (IDE). Шаблон плана испытаний.
<p>Ключ: СУ: Список упражнений; ШФ: Шаблон формы; СП: слайд презентации</p>	

Упражнения	Учебные заметки
Начальное упражнение (15 минут)	<ul style="list-style-type: none"> Представьте самостоятельную работу (см. «Рабочую тетрадь для оценки»), подробно описав основные критерии оценки, сроки и критерии сдачи.
Практическое оценочное упражнение (9 час 45 минут)	<ul style="list-style-type: none"> Используя Рабочую тетрадь для оценки, учащиеся выполняют модуль 1: Задания 1, 2 и 3. Учащиеся должны работать самостоятельно, выполняя данное задание. Учащиеся должны выполнять задание на оценку только под наблюдением.

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