ENGLISH FOR PROGRAMMERS

MASTER YOUR WORKFLOW IN ENGLISH:

REVIEW CODE - DISCUSS BUGS - AGILE PRACTICES

Levels: Pre-Intermediate / Intermediate / Upper-Intermediate / Advanced

www.speaktechenglish.com

ENGLISH FOR PROGRAMMERS ©

For any issues or questions, please contact:

contact@speaktechenglish.com

or

@speaktech.english

Foreword

Why we wrote this book

Our authors, Sophie and Tom, have been working in Tech and honing their skills as English instructors since 2017.

Both from the UK, Sophie worked as a Senior Data Analyst at a global leading financial information and analytics company and Tom worked as a Data Scientist at a marketing start-up.

Both TEFL certified and specialists in teaching Business English, they noticed two main issues for students:

- Very few English instructors have real-world corporate working experience, especially in STEM fields
- Lack of modern and natural learning resources that reflect how teams actually communicate at work

To address this, English for Programmers was created. This book's content focuses on natural, relevant and practical topics that accurately simulate situations you'll experience working as a programmer in today's global teams.

Who is this book for?

This book is perfect for English learners with job roles in Tech, such as:

- Software Engineers
- Data Engineers
- Data Scientists
- Data Analysts
- Web Developers
- Technical Project Managers
- QA Engineers
- Product Owners
- Cybersecurity Analysts
- System Analysts
- Network Engineers
- DevOps Engineers
- Database Administrators
- App Developers
- AI/Machine Learning Engineers
- Cloud Architects
- UI/UX Designers
- Full-stack Developers
- Embedded Systems Engineers
- Product Owners

What You Will Learn

On completion of this book, you will be able to:

Technical Discussions & Writing

- Use technical verbs to **accurately define tasks** and actions
- Write **commit messages** in the correct Git format
- Confidently **name** the **symbols** used when writing code
- Understand **vocabulary for syntax** and programming rules
- Differentiate between various **testing strategies**
- Write **professional guidelines**
- Showcase your **expertise** by using technical descriptors
- Correctly pronounce the names of **technical jargon**

Speaking

- Sound more natural and smooth when **asking questions**
- Use colloquial language in speaking to **give and accept feedback**
- **Avoid misunderstanding** when giving opinions on code
- Sound **more fluent** when speaking in past tense
- Give **neutral instructions or feedback** as a team lead to reduce blame
- Engage stakeholders when **presenting results**
- Identify patterns to **speak with** a more **natural rhythm**

Collaborating

- Listen to explanations of **complex topics** and extract key points
- Use a varied vocabulary for describing **problem solving**
- **Interpret** the meaning of **phrases** specific to the context of fixing bugs
- Use idioms to give **effective progress updates**
- **Guide** the flow and focus of **discussions** through language cues

How To Use

Each unit covers four sections:
 # vocabulary
 # grammar
 # pronunciation
 # listening

The unit introduction page will give you information on the topics you will learn, and provides a tick list to track your progress. You can also view the skills you will have gained after completing the unit.

Vocabulary, Grammar and Pronunciation

These sections contain:

- a page for **learning** - introducing and teaching the topic
- a page for **practising** - exercises to test your understanding

Listening

This section works a bit differently.

- Read the introduction and practice question, and listen to the audio to complete the exercise
- At the back of the book, you can find the transcripts. I know it's tempting, but try **not** to check this! Unless you really cannot understanding the listening, or want to review the answers at the end.

Answers

The answers to all exercises can be found at the back of the book

How To Use

Throughout the textbook, you will come across a range of icons and annotations to help your learning:

 **PLAY AUDIO FILE**

This opens the audio file in a new tab on Google Drive - don't worry, you **don't need** a Google account to play it.

This gives you more explanation, like a definition or example, on specific words and phrases

example answer

For some exercises, the first question has been completed for you and is shown in blue writing.

Important things to watch out for

Helpful tips and advice

Extra hints on the exercise questions

Contents

1

Implementing Code

vocabulary action verbs
grammar imperative present tense
pronunciation keyboard symbols
listening syntax

Tick off your progress!

AFTER THIS UNIT, YOU WILL BE ABLE TO:

Use technical verbs to **accurately define tasks** and actions

Write **commit messages** in the correct Git format

Confidently **name** the **symbols** used when writing code

Understand **vocabulary** for **syntax** and programming rules

 # vocabulary

Action Verbs

words that express an activity or action

As a programmer, the use of action verbs helps to define specific tasks and actions in the code development process.

Take a look at the example sentences below. Consider how the verbs in **bold** can be used instead of the phrases on the right hand side.

He **optimised** the queries to improve the response time.	**(improved)**
Can you **implement** the new feature we discussed yesterday?	**(put into action)**
The team will **integrate** a third-party API to get real-time data.	**(combine)**
As our user base grows, we'll need to **scale** our infrastructure.	**(increase capacity)**
Have you had a chance to **refactor** the code yet?	**(change)**
The process is taking too long. How can we **streamline** it?	**(simplify)**
Let's **execute** the script before we go for lunch.	**(run)**
The settings haven't been **configured** yet.	**(set up)**

Note: the spelling of optimise
British English '-ise' vs. American English '-ize'
Other examples include organise, prioritise, etc...

USE ACTION VERBS TO...

- break down complex processes into actionable steps
- provide clear instruction on the task that needs to be performed

vocabulary

Action Verbs

Exercise 1A

You have been sent a list of issues that have been identified in the code development process

i) answer each concern with a resolution using an action verb + them/it

optimise	**implement**	**refactor**	**execute**
integrate	**scale**	**streamline**	**configure**

Problems	Resolutions
1. The parameters haven't been set	Let's *configure them*
2. Changes need to be made to the code base	Let's _____
3. The systems should work together	Let's _____
4. The scheduled tasks didn't run	Let's _____
5. Our workflow is too complicated	Let's _____
6. The pipeline should be more efficient	Let's _____
7. We need a user authentication process	Let's _____
8. The database has reached it's capacity limit	Let's _____

grammar

Imperative Present Tense

Commit messages detail the changes made to a codebase, providing context not only for yourself but also for future developers.

For readability and consistency in commit messages within a team, Git recommends using the Imperative Present Tense

the imperative mood:

- **tells someone to do something**
- **uses the base form of the verb and usually no explicit subject**

e.g. **commands**
requests
instructions

"Alexa, set a timer for 5 minutes."
"Please call me."
"Submit your reports by Friday."

When writing commit messages, think of them as instructions to the version control system and other developers.

TOP TIP
Your message should describe what applying the commit will do, not what you did

Think,
"If I apply this commit, I will _(insert commit message)_"

Recommended	Not Recommended
Add new feature for user authentication Resolve issue with data validation	Added a new feature for user authentication Resolved the issue with data validation

when writing imperative sentences, we can omit articles (a/an/the)

 # grammar

Imperative Present Tense

Exercise 1B

Your team follows the version control strategy recommended by Git, where each commit message is expected to be in the imperative present tense.

i) Rewrite the following commit messages to use the imperative present tense. Remember you can omit articles.

1. Changed the colour scheme of the homepage

_____

2. Updating the library dependencies

_____

3. Implemented a new algorithm for sorting

_____

4. Fixed a bug in the login module

_____

5. Adding new features to the dashboard

_____

6. Refactored the code for better readability

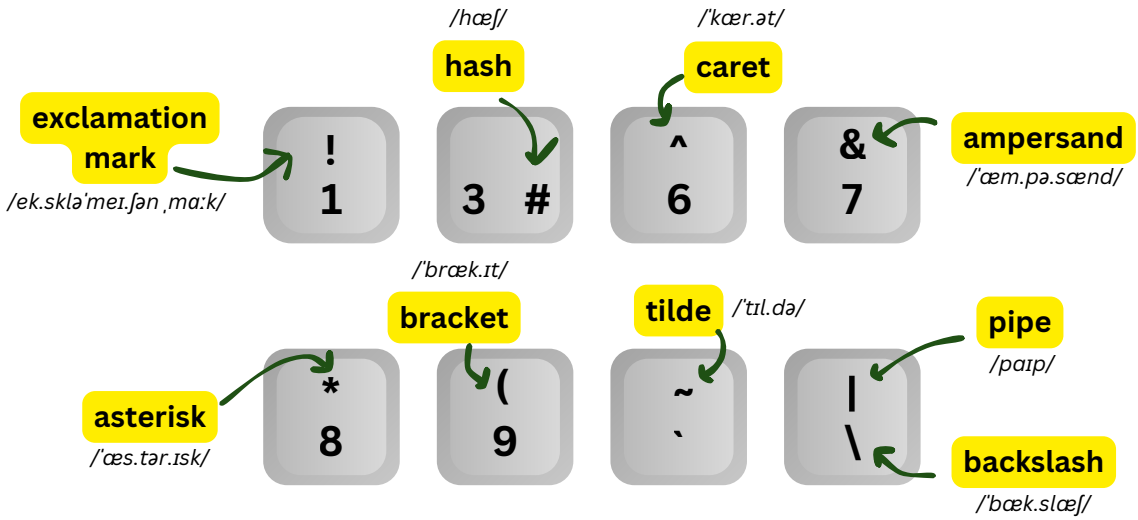
_____

pronunciation

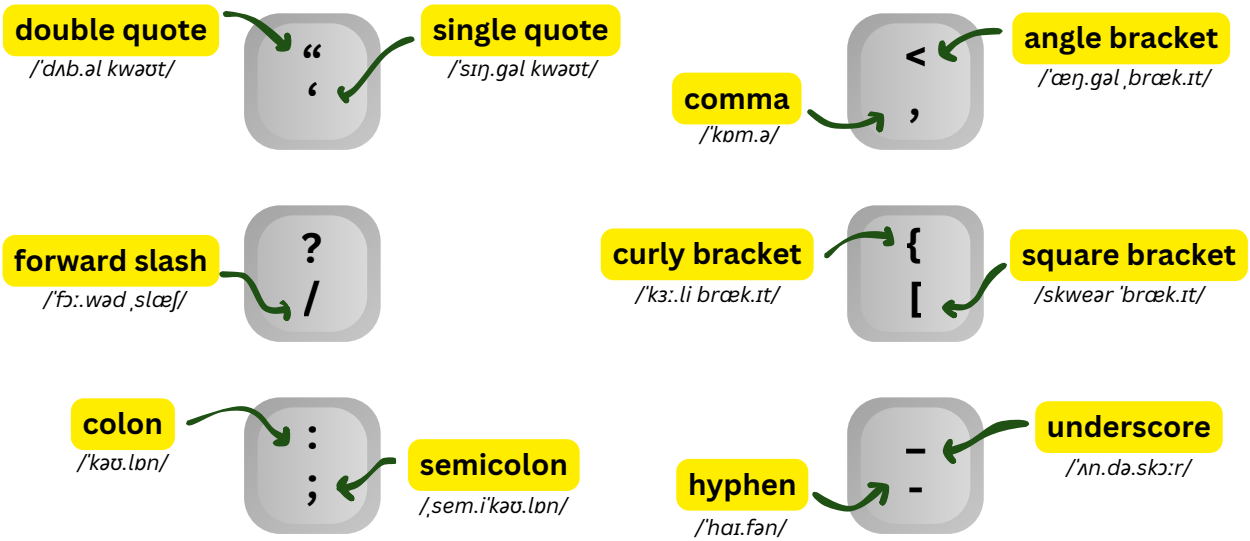
Keyboard Symbols

Imagine you're on a call with a team mate, developing some code together.
As they review your work and offer advice on what needs changing, they suggest,

" Can you try replacing the **asterisk** with an **ampersand**
and adding a **tilde** after the **pipe?**"



ⓘ *Note: this is in British English, names may differ in American English*



pronunciation

Keyboard Symbols

Exercise 1C

Compare the original and corrected code in the below snippets

i) Describe the changes made using the names of the keyboard symbols

1

```
# original
if result != 0:
    print("Result is zero.")

# corrected
if result == 0:
    print("Result is zero.")
```

replaced the exclamation mark with an equals

Specify if it's an **open** bracket or **close** bracket

2

```
# original
print("Process complete."

# corrected
print("Process complete.")
```

inserted missing closing bracket

3

```
# original
result = a | b

# corrected
result = a & b
```

replaced the pipe with an ampersan

4

```
# original
result = num1 ^ num2

# corrected
result = num1 * num
```

Replaced the carat with the asterisk

Exercise 1D

Fill in the gaps with the name of the correct keyboard symbol

1. Kebab case is a naming convention where all letters are lowercase and words are separated by Hyphens, e.g. my-variable
2. Snake case is a naming convention where all letters are lowercase and words are separated by Underscore, e.g. my_variable
3. Many programming languages use single quotes or double quotes to denote strings, e.g. "This is a string."
4. HTML tags are enclosed in angle brackets, e.g.

listening

Syntax

rules defining the structure of the symbols, punctuation and words of a programming language

Exercise 1E

Your friend is telling you about the syntax of the new language they have written

i) **Listen to the audio**
ii) **Answer the multiple choice questions, giving a reason for your choice**

 **PLAY AUDIO FILE**

1. **Which symbol is used to represent a comment?**
   a. *****
   b. **&**
   c. **#**
   d. **;**
2. **Which statement best defines the rules on indentation?**
   a. It must be strictly followed
   b. It's optional
   c. It's not possible to indent code
   d. Only specific code blocks should be indented
3. **Which of the function names follows the naming conventions?**
   a. duplicationperform
   b. duplication123
   c. 123performduplication
   d. performduplication
4. **Which variable would be treated the same as the variable: "AGE":**
   a. Age
   b. age
   c. Both of the above
   d. None of the above
5. **Which variable name is NOT valid?**
   a. new_data
   b. POPULATION
   c. over85s
   d. increaseVersion

NEED SOME HELP...
Click here to check the transcript - but only if you reaaally need it!

2

Code Review & Testing

vocabulary noun phrases
grammar parallel structure
pronunciation connected speech
listening code review

Tick off your progress!

AFTER THIS UNIT, YOU WILL BE ABLE TO:

Differentiate between various **testing strategies**

Write **professional guidelines**

Sound more natural and smooth when **asking questions**

Use colloquial language in speaking to **give and accept feedback**

vocabulary

Noun Phrases

Testing is an important phase in software development to check that software meets certain standards and user requirements.

Use these noun phrases to demonstrate fluency in technical English:

time box	**an allocated period of time for completing a task**	• set a maximum of 15 minutes for code review • allocate a 2-hour time box for regression testing
stress test	**a method to assess a system's performance under heavy loads**	• simulate 1000 users accessing the login page at the same time • increase server load to test response time under heavy load
sanity check	**a quick check to verify that something is as expected**	• does the 'home' button redirect to the homepage? • are the units of the output value correct?
ad hoc test	**a test performed without predefined test cases or plans**	• input unexpected characters into a search bar • interrupt a process mid-flow and check error logs
edge case	**a problem that only happens in extreme situations**	• input a birth date of 01/01/1900 • upload an empty, 0-byte file

SOME NOUN PHRASES CAN BE USED AS VERBS

e.g. "Can you sanity check my email before I send it? I want to make sure there aren't any errors."

OR

"We have a lot to do today. Let's time box this meeting so we stay on schedule."

vocabulary

Noun Phrases

Exercise 2A

Fill in the blanks with an appropriate noun phrase from the list

time box **stress test** **sanity check**

edge case **ad hoc test**

1. How would the weather forecasting app handle an _____ of, for example, 100 degrees celsius?
2. Yesterday, while doing _____, I came across some unexpected behaviour when I was randomly interacting with the system.
3. The dev team conducted a _____ by simulating a large number of users accessing the website at the same time.
4. Hey, I made a few changes to the codebase. Can you do a quick _____ before I start testing?
5. I'd like each team member to set a _____ of one hour for code reviews please. They shouldn't be taking any longer than this.

BE CAREFUL: sanity check vs ad hoc test
sanity check: similar to performing a review
ad hoc test: exploring issues using the tester's knowledge

 # grammar

Parallel Structure

What is parallel structure?

➞ Using the same grammatical structure for two or more clauses in a sentence

Let's jump straight in with an example:

> "In our coding guidelines, we emphasise **writing** clear comments, **to follow** naming conventions, and **maintain** consistent indentation."

What do you notice about the **verbs**?

writing *= gerund form*
to follow *= infinitive form*
maintain *= base form*

❗ **MIXED VERB FORMS**

To achieve parallel structure, **change the verbs to the same form**.

Since this example begins with 'emphasise', which is typically followed by a noun or gerund, we will choose the **-ing form:**

> "In our coding guidelines, we emphasise **writing** clear comments, **following** naming conventions, and **maintaining** consistent indentation."

Benefits of Parallel Structure:
- more professional
- more effective
- easier to read and follow

WHERE TO USE...

You can apply this technique when writing:
- documentation
- code comments
- best practices guidelines

grammar

Parallel Structure

Exercise 2B

Written below are guidelines set by your team which should be followed when reviewing each other's code

i) Revise the checklist to ensure parallel structure by changing the verbs to a suitable form

Code Review Checklist

1. Verifying that functions are documented adequately
2. Checked for proper error handling
3. Ensure that indentation is consistent
4. Avoiding duplicated code
5. To write modular functions

Revised Version:

Code Review Checklist

1. _____
2. _____
3. _____
4. _____
5. _____

NEED SOME HELP...
Think about what verb form to use.
Since this is a checklist, consider what we would use to give clear commands or requests

 # pronunciation

Connected Speech

When we talk in everyday conversations, our words shouldn't stand alone.

Instead, some sounds, words and phrases are merged together in what's called connected speech. It's a natural rhythm and flow that make conversations sound more smooth.

Let's take a closer look at three different techniques that you can use while asking questions:

ASSIMILATION joining two sounds to make a new sound	could you	coujoo /ˈkʊdʒu/
REDUCTION shortening or removing particular sounds	who is	hooz /huːz/
LINKING joining the final sound of one word to the first sound of the next word, without a pause	how about	how wabout /haʊbaʊt/

/d/ + /y/ = /dʒ/

Using connected speech in testing & QA related questions:

coujoo
Could you help me with testing today?

/d/ + /y/ = /dʒ/

wheredja
Where did you log that?

doesyer
Does your script check for compatibility errors?

/z/ + /y/ = /ʒ/

wenna'we
When are we meeting?

linking / reduction

whadja
What did you find during QA?

d/ + /j/ = /dʒ/

pronunciation

Connected Speech

Exercise 2C

Read the following sentences

i) Underline the places where connected speech techniques may be used

Example:
Did you run the test cases? *didjarunthuh test cases?*

1. When is the next release scheduled?
2. Who is doing the testing?
3. Has your framework been stress tested?

Exercise 2D

Listen to the audio file. The speaker will read the below sentences twice.

i) Identify which time, A or B, the speaker uses connected speech

 **PLAY AUDIO FILE**

1. Have you had a chance to review the scripts?	**A**	**B**
2. How is it going?	**A**	**B**
3. What are you working on today?	**A**	**B**

listening

Code Review

In the audio, Tom has called Sophie to share his findings after reviewing her code for an ETL pipeline.

 **PLAY AUDIO FILE**

Exercise 2E

Listen carefully to the audio and rewrite the underlined section of the quotes below with the phrase that was used in the dialogue

Example:
It did need reorganizing.
It did need a good tidy up

1. I've got a few points I'd like to discuss.

2. I like the changes you made to the data loading module.

3. Let me just open the code on my screen.

4. I'll try adding some try-except blocks here.

5. Overall, it's very well structured.

NEED SOME HELP...
Click here to check the transcript - but only if you reaaally need it!

3

Discussing Code

vocabulary modifiers
grammar placement of modifiers
pronunciation -ed verbs
listening case study: ChatGPT

Tick off your progress!

AFTER THIS UNIT, YOU WILL BE ABLE TO:

Showcase your **expertise** by using technical descriptors

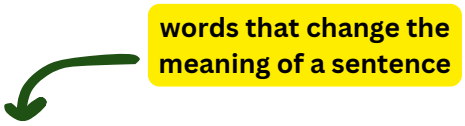
Avoid misunderstanding when giving opinions on code

Sound **more fluent** when speaking in past tense

Listen to explanations of **complex topics** and extract key points

vocabulary

Modifiers



When discussing code, modifiers can be extremely useful for making your feedback constructive and precise.

For example, a colleague asks:

What do you think of the website?

Response 1

It's good! When I click something it loads very quickly and I can navigate it easily without any instructions

Response 2

It's good! It's very responsive and can be navigated intuitively

Which response is better?
By using technical adjectives and adverbs, Response 2 sounds more clear and professional

Let's explore other modifiers you can use to give your opinion on technical topics:

scalable	**robust**	**consistent**
user-friendly	**reusable**	**seamlessly**

Attempt the exercises on the next page to find their definitions and check your understanding.

USE MODIFIERS TO...
focus attention and set accurate expectations

vocabulary

Modifiers

Exercise 3A

Rewrite the underlined section of the sentences using a modifier from the list

scalable	**robust**	**consistent**
user-friendly	**reusable**	**seamlessly**

Example:

I like the interface of GitHub; it's very easy to use and understand.

I like the interface of GitHub; it's very user-friendly

1. Developers should follow the same coding practices.

2. Can you explain why a strong code review process is important?

3. New features should be integrated smoothly and without disruptions.

4. I want you to focus on building components that can be used many times.

5. Our infrastructure needs to be able to be made larger to handle an increase in users.

grammar

Placement of Modifiers

known as a misplaced modifier

When a modifier isn't used in the correct position, it can make the sentence confusing for a reader/listener.

noun

incorrect: "The algorithm solved **quickly** the problem."

verb

Should '*quickly*' modify the verb or the noun?

correct: "The algorithm solved the problem **quickly**."
OR
correct: "The algorithm **quickly** solved the problem."

Adverbs modify **verbs**, **adjectives** or other **adverbs**.
They can be placed in different positions depending on what the intended emphasis of the sentence is.

Adjectives usually come before the noun they modify.

"We offer **scalable** solutions."

noun

TOP TIP
Place modifiers as close as possible to the word they modify to avoid confusion

Consider this example, taken from a development team's best practices manual:

> **"Our code review process ensures that every line of code is checked rigorously. We efficiently conduct detailed reviews and embrace continuous improvement to deliver functional and maintainable code."**

describing how an action is performed

Adverbs of manner typically come before the main verb they modify or at the end of the clause/sentence.

Therefore, '**efficiently**' could be repositioned:
"We conduct detailed reviews **efficiently** and embrace..."

grammar

Placement of Modifiers

Exercise 3B

For each question, add the given modifier into the correct position in the sentence

Example:

maintainable: It's important to write code.

It's important to write maintainable code.

1. **briefly**: The comment should describe what each function does.

2. **accurately**: Using a version control system helps to track changes.

3. **critical**: Joe mentioned several issues that need immediate attention.

4. **constructive**: I want to thank the team for providing feedback.

5. **versatile**: I spoke with the team and they liked our framework.

 # pronunciation

-ed Verbs

Verbs ending in -ed can be pronounced in three different ways: **/t/**, **/d/**, or **/ɪd/**.

How can we determine which pronunciation to use?
→ look at the final sound of the verb's base form
not the letter!

It goes without saying, using the correct pronunciation is important to ensure your team understands and interprets your words accurately.

Here are some general rules:

e.g. p, k, s, sh, ch, f

e.g. b, g, v, th, vowels

ending sound of base form of verb	pronunciation of -ed	example	extra syllable?
unvoiced	/t/	pu**sh**ed	No
voiced silent -e	/d/	debug**g**ed resolv**e**d	
/t/ /d/	/ɪd/	adop**t**ed downloa**d**ed	Yes

Note: unfortunately, you might meet some exceptions to these rules

What is voiced vs. unvoiced?

A **voiced sound** causes your vocal cords to vibrate when you speak, and the opposite is known as **unvoiced** or **voiceless**.

You can tell the difference by putting your hand on your throat when you speak. For example, try saying /m/ (voiced) and /f/ (unvoiced).

PLAY AUDIO FILE

Listen to the audio to check the pronunciation of the examples in the table above

 # pronunciation

-ed Verbs

Exercise 3C

Read the following sentences and identify the -ed verbs

i) Determine whether the verb is pronounced with /t/, /d/ or /ɪd/ ending

What day was the latest version deployed? **/d/** *since deploy = /dɪˈplɔɪ/ and /ɔɪ/ is voiced*

1. Code reviews are now faster because we've automated style checks.
2. Mapped diagrams visually show how different modules interact in our codebase.
3. How quickly were these changes processed?
4. I've evaluated the performance metrics of the new feature.
5. He's proud of the solution he coded for this issue.
6. Let me know when you have the refined version.

listening

Case Study: ChatGPT

Exercise 3D

Listen carefully to the audio describing how ChatGPT works and answer the following comprehension questions

 **PLAY AUDIO FILE**

1. What does GPT stand for?

_____

2. In what manner does GPT process and understand text data?

_____

3. What was the first stage of building ChatGPT?

_____

4. What's the name of the process that teaches the model to predict language patterns?

_____

5. What's the name of the iterative refinement process that enables the model to be high-performing and adaptable?

_____

6. ChatGPT opens up new possibilities in what three areas?

_____

NEED SOME HELP...
Click here to check the transcript - but only if you reaaally need it!

4

Bug Fixing

vocabulary phrasal verbs
grammar passive voice
pronunciation word stress
listening daily scrum

Tick off your progress!

AFTER THIS UNIT, YOU WILL BE ABLE TO:

Use a varied vocabulary for describing **problem solving**

Give **neutral instructions** or feedback as a team lead to reduce blame

Identify patterns to speak with a more **natural rhythm**

Interpret the meaning of **phrases** specific to the context of fixing bugs

 # vocabulary

Phrasal Verbs

verb + particle (adverb / preposition)

Why use phrasal verbs?

- describes actions in a more detailed way
- gives a natural and conversational tone
- provides a varied vocabulary

For example,

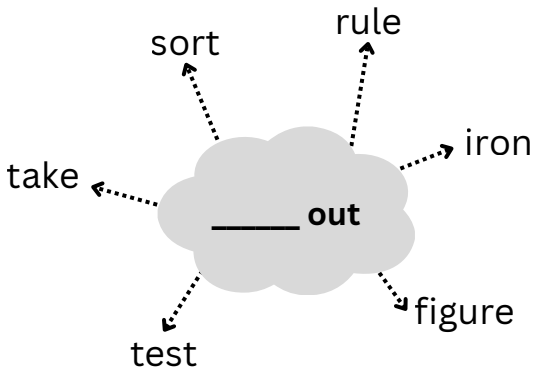
phase out

"I think we should ~~gradually stop using~~ this feature."

Phrasal verbs are commonly used in discussions related to bugs, such as in Scrum meetings or other progress updates, to accurately explain the status of the issue.

Let's look at some examples that you can use in communication at work.

Specifically, phrasal verbs ending with the particle 'out'.



THINGS TO CONSIDER

1

All of the above phrasal verbs are **separable**

i.e. the object can be placed between the verb and the particle OR after the particle

Did you **test** the software **out**?

Did you **test out** the software?

2

Phrasal verbs can have multiple meanings depending on the context

rejected

He **turned down** the new job opportunity.

The music was too loud, so he **turned** it **down**.

decreased the volume

3

Phrasal verbs are most commonly used in spoken English and informal writing

In official or academic writing, it's best to use their formal equivalent

We will ~~**look into**~~ the issue

investigate

vocabulary

Phrasal Verbs

Exercise 4A

Read the phrasal verbs

i) **Match each one with it's definition**
ii) **Find the example sentence that can be rewritten using the phrasal verb**

	Phrasal Verb	Definition	Example
1	sort out	find the answer to something	We still have a few problems that need to be resolved
2	rule out	reorganise	Try removing the function and running it again
3	iron out	resolve issues	I'm trying to determine why the output is incorrect
4	take out	check if something is working	Can you rearrange the code? It's a bit messy
5	test out	decide something isn't an option	We can eliminate network issues as the cause of this bug
6	figure out	remove something	Did you try the new feature?

grammar

Passive Voice

object + be verb + past participle

There is a popular expression in English, “it’s not what you say, it’s how you say it.”

Passive voice can be used to soften feedback or place emphasis on a task that needs to be performed.

STEPS:

1. Identify the subject, verb and object in the original sentence
2. Move the object to become the subject
3. Use the appropriate form of ‘to be’
4. Follow this by the past participle of the main verb

S V O

Original: "You need to fix this bug"

Passive: "This bug needs to be fixed"

(2) (3) (4)

USE PASSIVE VOICE TO...

shift the focus from who is doing the action to the action itself

Example 1: Active Voice

Today

Tom 09:01 AM

Morning All,
We had a meeting yesterday to investigate the recent system crash.

Here are the notes:

- Joe found a critical bug in the login module
- I deleted a function from the shared library and this caused something to fail
- I fixed the alignment issue

Example 2: Passive Voice

Today

Tom 09:01 AM

Morning All,
We had a meeting yesterday to investigate the recent system crash.

Here are the notes:

- A critical bug was found in the login module
- A function was deleted from the shared library and this caused something to fail
- The alignment issue was fixed

the focus is on the actions

Benefits of Passive Voice:

- avoids direct blame
- encourages a constructive, problem solving approach
- maintains a professional tone

grammar

Passive Voice

Exercise 4B

As a Tech Team Lead, you often need to provide guidance and instructions to your team, as well as stay up to date with progress

i) Adjust the given sentences to passive voice and remove direct blame

Example:

The issue occurred because she didn't update the settings on the server.

The issue occurred because the settings on the server weren't updated

1. You should implement the new feature according to the specifications.

_____

2. Did Michael inform the team about the upcoming fixes to the codebase?

_____

3. You need to show the devs how to tackle those security issues we found.

_____

4. Did you not detect this issue before it went into production?

_____

5. Why didn't anyone catch this bug during testing?

_____

NEED SOME HELP...

Check the hints and steps on the previous page for help with the structure

pronunciation

Word Stress

Imagine you are speaking to someone on a call, but the connection is bad.
You only hear the first two syllables of a word.. **pho·to...**

*Did they say **photograph** or **photographer**?*

which part of the word to emphasise

With the beauty of word stress, you will know immediately, because you will either hear:

PHO·to.... OR **pho·TO....**
PHO·to·graph *pho·TO·graph·er*

TOP TIP
Recognising word stress will help you understand those who speak very fast

Let's look at some examples of stress patterns in words associated with writing and debugging code.

Some words maintain the same form as nouns and verbs.

Look at the examples in the table.

What do you notice?

Word	Noun	Verb
increase	IN·crease	in·CREASE
update	UP·date	up·DATE
record	RE·cord	re·CORD
upgrade	UP·grade	up·GRADE

stress first syllable

stress second syllable

Word	Noun
execution	ex·e·CU·tion
integration	in·te·GRA·tion
version	VER·sion
regression	re·GRES·sion

We can also see a pattern in words ending in -tion or -sion

stress second-from-last syllable

pronunciation

Word Stress

Exercise 4C

Read the below sentences. Using the rules on the previous page,

i) **Identify which syllable should be stressed for the bold words**
ii) **Verify your pronunciation using the audio file**

e.g. There's a **conflict** between these two branches in Git.

1. If the user doesn't choose a language, the system **defaults** to English.
2. This error might have occurred during the **transition** to the new software.
3. It looks like you forgot to **import** the correct modules.
4. If you try to merge the branches now, they might **conflict**.
5. The logs show the **deletion** of an important module.
6. The tool provides an **extension** that allows developers to analyze memory usage more effectively.
7. If the user doesn't choose a language, the **default** is English.
8. A ZeroDivisionError is an example of an **exception** in Python.

 **PLAY AUDIO FILE**

listening

Daily Scrum

In Agile methodology, a daily scrum or stand-up is an opportunity for team members to discuss their plans for the day, including information on any bugs they've found and how they will approach fixing them.

Exercise 4D

Listen to the audio which contains 5 clips from a progress update meeting

i) Identify the vocabulary used in each mentioned clip that matches the given definition

 **PLAY AUDIO FILE**

	Definition	Vocabulary	Clip
noun	a method used to avoid a problem but doesn't actually solve it		
noun phrase	problems arising from past decisions		**1**
noun	the person intended to use a product	*end-user*	
noun	range/size/amount		**2**
phrasal verb	to experience a problem		
phrasal verb	to meet or find something by chance		**3**
idiom	ready		
adjective	easy to do or understand		**4**
verb	cause something to happen		
verb	suddenly failing		**5**
verb	create something again		

NEED SOME HELP...
Click here to check the transcript - but only if you reaaally need it!

5

Collaboration & Meetings

vocabulary idioms
grammar active voice
pronunciation common mispronunciations
listening retrospective meeting

Tick off your progress!

AFTER THIS UNIT, YOU WILL BE ABLE TO:

Use idioms to give **effective progress updates**

Engage stakeholders when presenting results

Correctly pronounce the names of **technical jargon**

Guide the flow and focus of **discussions** through language cues

vocabulary

Idioms

As a programmer, whether you like it or not, participating in meetings is still an important part of the job!

Here, we're going to focus on meetings where you share your progress.
This could be:

- Daily Scrum/Stand-Up
- Demo or Show-and-Tell
- Retrospectives
- Project Updates

Using idioms in these types of meetings, especially where time is limited, helps communicate progress quickly without getting caught up in lengthy explanations.

Idioms are often deeply rooted in a language's culture and history. If you can master how and when to use them, you can demonstrate a deeper understanding of the language and connect better with native team members.

	**in the pipeline**	**currently being developed or planned**	We have some exciting features in the pipeline that will be ready for deployment in the next sprint.
	**on the back burner**	**temporarily not dealing with something**	This project has been put on the back burner while the client revises their request.
	**put out fires**	**to resolve urgent and critical issues**	The team have been working hard this week, putting out fires related to integration challenges.
	**down the rabbit hole**	**deeply involved in a complex problem**	While initially investigating a performance issue, we started going down a rabbit hole and discovered database errors.
	**up to speed**	**fully informed or up to date**	Let me bring you up to speed with the latest progress.

vocabulary

Idioms

Exercise 5A

Where do these idioms come from? Listed below is the supposed origin of each

i) Match the story with the correct idiom

Inspired by Lewis Carroll's novel, referring to Alice's unexpected journey in to Wonderland through a rabbit hole	Originally referred to a ship achieving full or optimal speed for navigation	Has a literal origin in manufacturing, where items are in a development line for production	Comes from the idea of firefighters rushing to extinguish fires to prevent further damage	A cook will move pans to the back of the stove so he can focus on dishes that need immediate attention

Exercise 5B

Fill in the gap with the idiom that accurately summarises each sentence

Example:

I spent hours researching that topic - it wasn‘t as simple as I thought.

Researching that topic *took me down a rabbit hole*

1. Sarah’s been on holiday for 2 weeks and has missed lots of important updates.

   Sarah needs to _____ when she returns.

2. Due to budget constraints, we’ve decided to postpone the development of this module. The project has been put _____ for now.

3. We’re working on the upgrade and it’s scheduled to be released next month.

   The upgrade is _____.

grammar

Active Voice

subject + verb + object

Active voice is used to place focus on who is doing the action.

When presenting to stakeholders, using active voice will clearly show who took responsibility for a task or achievement.

(1)

Original: "A new feature was implemented by the dev team."

Active Voice: "The dev team implemented a new feature."

(2)

STEPS:

1. Identify who is performing the action (the doer)*
2. Restructure to bring the doer before the verb

Benefits of Active Voice:

- more direct and easy to follow
- engages the audience
- helps to clearly explain complex technical concepts to non-technical stakeholders

Original: "The report was submitted."

Active Voice: "They submitted the report."

*Sometimes, the doer of the action is not stated.

e.g. someone / they / I

Here, you can use a pronoun, or a general term related to the context

e.g. the analysts

ACTIVE VOICE IN PRESENTATIONS

- Use strong and descriptive verbs
- Highlight key players
- Focus on results and achievements

grammar

Active Voice

Exercise 5C

Imagine you are presenting the progress of a data analytics project to stakeholders

i) Restructure the following sentences into active voice (if they are not already)

1. Key trends were identified during the data analysis.

_____

2. The analysts are preparing a detailed report on the results.

_____

3. An interactive dashboard was created by Joe for visualising the data.

_____

4. Based on the analysis, we provided recommendations for optimisation.

_____

5. When analysing large datasets, valuable insights are provided by tools like Tableau.

_____

_____

6. Thanks for your questions. Our team will investigate these as soon as possible.

_____

NEED SOME HELP...

- Look for a 'by' phrase to find *the doer*
- Move *the doer* to the start of the sentence

 # pronunciation

Common Mispronunciations

Is it **SQL** (ess-que-el) or **Sequel**?

special language

When it comes to pronouncing technical jargon in English, some can be debated, like SQL, and others are just frequently mispronounced.

Let's take a look at some examples

Jargon	Pronunciation	IPA
API	AY·pee·eye	/ˈeɪ.pi.aɪ/
WiFi	WHY·fy	/ˈwaɪ.faɪ/
JSON	JAY·sun	/ˈdʒeɪ.sən/
CI/CD	SEE·eye·SEE·dee	/si ˌaɪ ˌsiː ˈdiː/
Linux	LIN-ooks	/ˈlɪnʊks/
Ubuntu	oo-BOON-too	/ʊˈbuːntuː/
Cache	kash	/kæʃ/

PLAY AUDIO FILE

Practise the pronunciation of the tech terms and use the audio to verify your pronunciation

make sure you are placing the stress on the correct syllables

 # pronunciation

Common Mispronunciations

Exercise 5D

Listen to the audio file. The speaker will read the below sentences twice.

i) Identify which version of the sentence, A or B, was pronounced correctly

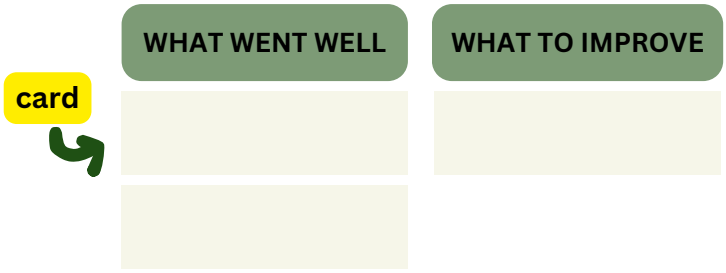
1. Are you having problems with the API?	**A**	**B**
2. Sorry, I'm back. My WiFi dropped.	**A**	**B**
3. I've got the file but it's not a JSON.	**A**	**B**
4. CI/CD is an important part of our development process.	**A**	**B**
5. Do you have experience with Linux?	**A**	**B**
6. I need some help setting up an Ubuntu environment.	**A**	**B**
7. Try clearing the cache and then run it again.	**A**	**B**

 **PLAY AUDIO FILE**

listening

Retrospective Meeting

Retros occur at the end of a project for a development team to reflect on what went well, what didn't, and how processes can be improved going forwards.



Exercise 5E

Listen to the audio file, which is a clip from a retrospective meeting, and answer the following questions

PLAY AUDIO FILE

1. What **initiative** was considered successful with unanimous agreement?

_____

2. What does the phrase "**tie up loose ends**" mean?

_____

3. What **transition phrase** is used to shift the focus to the next topic?

_____

4. Which **two phrases** are used to highlight issues with the depth and range of the unit tests?

_____

5. What phrase is used to **bring attention back** to the topic of unit testing?

_____

6. What phrase is used twice to **suggest potential solutions** in the form of a question?

_____

7. **True or False?** A meeting will be held on Thursday or Friday of the next week for developers to discuss improvements to unit testing.

_____

_____

NEED SOME HELP...
Click here to check the transcript - but only if you reaaally need it!

Answers & Transcripts

 # UNIT 1

Answers

Exercise 1A

1. configure them
2. refactor it
3. integrate them
4. execute them
5. streamline it
6. optimise it
7. implement it
8. scale it

Exercise 1B

1. Change colour scheme of homepage
2. Update library dependencies
3. Implement new algorithm for sorting
4. Fix bug in login module
5. Add new features to dashboard
6. Refactor code for better readability

Exercise 1C

1. replaced the exclamation mark with an equals
2. added the missing close bracket to the end of the print statement
3. switched the pipe for an ampersand
4. switched the caret for an asterisk

Exercise 1D

1. hyphens
2. underscores
3. single quotes or double doubles quotes
4. angle brackets

Exercise 1E

1. B (comments denoted using ampersand)
2. B (indentation is not mandatory)
3. D (function names must start with a verb)
4. C (variables are not case sensitive so AGE=Age=age)
5. A (variable names can only contain a-z or 0-9)

 # listening

Syntax

Transcript for Exercise 1E

Let me explain the syntax of my programming language.

So firstly, variable names aren't case sensitive and can only contain alphanumeric characters.
Secondly, comments can be denoted using an ampersand at the beginning.
Thirdly, indentation is not mandatory but it is encouraged for readability.
Fourthly, function names must start with a verb and be descriptive of their purpose.
And finally then, mathematical symbols are not allowed to be used

Click to go back to exercise 1E

 # UNIT 2

Answers

Exercise 2A

1. edge case
2. ad hoc tests
3. stress test
4. sanity check
5. time box

Exercise 2B

1. Verify that functions are documented adequately
2. Check for proper error handling
3. Ensure that indentation is consistent
4. Avoid duplicated code
5. Write modular functions

Exercise 2C

1. whensthuh next release scheduled?
2. whoozdointhuh testing?
3. Hazya framework been stress tested?

Exercise 2D

1. A
2. B
3. B

Exercise 2E

1. I've got a few points I'd like to go over
2. I like how you refactored the data loading module
3. Let me just pull it up
4. I'l have a go at adding some try-except blocks
5. Overall, it's looking really solid

listening

Code Review

Click to go back to exercise 2E

Transcript for Exercise 2E

Tom: Hey Sophie!

Sophie: Hi! How are you?

Tom: All good, all good. So I've reviewed the changes that you made to the ETL pipeline. Overall, it looks great, but I've got a few points I'd like to go over.

Sophie: Sure ok, let me just pull it up. Ok I'm ready, go ahead

Tom: Firstly, in the data transformation phase, I noticed a nested loop structure that might impact performance when we go to handle large datasets. Have you thought about optimising this bit?

Sophie: Yeah, I see what you mean. So, instead of a loop, what are you thinking?

Tom: I was thinking you could use a list comprehension for that part.

Sophie: Ok sure, let me go back and review it and I'll give that a go.

Tom: In terms of error handling, I noticed some areas where exceptions aren't being caught properly. These are crucial since it means it's not going to crash the entire system.

Sophie: Ok good point. I'll have a go at adding some try-except blocks here and then I'll go over the error logging to make sure we've got details if there are any exceptions.

Tom: Sounds good. On a positive note, I really like how you refactored the data loading module. It's much cleaner and easier to follow now.

Sophie: Oh yeah it was a bit of a mess to be honest so it did need a good tidy up. Any other points?

Tom: Nope I think that's everything, overall, it's looking really solid. I'll leave some comments on the code with everything that I've mentioned for improvement, but great work overall. Well done.

Sophie: Perfect. Thank you. Thanks for the feedback. I'll get started on those and then I'll let you know when it's good to go.

Tom: Alright. Thanks! Have a great day. Bye.

Sophie: Yep and you! Bye.

 # UNIT 3

Answers

Exercise 3A

1. consistent
2. robust
3. seamlessly
4. are reusable
5. scalable

Exercise 3B

1. The comment should briefly describe what each function does.
2. Using a version control system helps to accurately track changes.
3. Joe mentioned several critical issues that need immediate attention.
4. I want to thank the team for providing constructive feedback.
5. I spoke with the team and they liked our versatile framework.

Exercise 3C

1. automated - /ɪd/
2. mapped - /t/
3. processed - /d/
4. evaluated - /ɪd/
5. coded - /ɪd/
6. refined - /d/

Exercise 3D:

1. Generative Pre-Trained Transformer
2. Hierarchical
3. Collecting huge amounts of textual data
4. Pre-training
5. Fine-tuning
6. Communication, assistance, creativity in programming

 # listening

Case Study: ChatGPT

Transcript for Exercise 3D

ChatGPT, is a revolutionary language model developed by OpenAI. At its core, ChatGPT was constructed utilizing a sophisticated deep learning architecture known as the Generative Pre-trained Transformer (GPT). This architecture comprises numerous layers of neural networks that process and understand text data in a hierarchical manner, allowing it to generate consistent and contextually relevant responses.

The construction of ChatGPT began with the collection of huge amounts of textual data from diverse sources such as books, articles, and websites. This data served as the foundation for training the model to understand the intricacies of human language. Through a process called pre-training, the model was exposed to this data and learned to predict the next word in a sequence of text, effectively capturing the underlying patterns and nuances of language usage.

Following pre-training, ChatGPT underwent fine-tuning, a process where the model's parameters were adjusted and refined to better suit specific tasks or domains. Fine-tuning allowed ChatGPT to specialize in various applications, ranging from answering questions and providing assistance to engaging in natural conversation. This iterative refinement process played a crucial role in enhancing the model's performance and adaptability across different contexts.

Overall, the development of ChatGPT represents the combination of cutting-edge research in artificial intelligence and machine learning. By leveraging the power of deep learning techniques and large-scale data processing, ChatGPT has emerged as a groundbreaking tool for understanding and generating human-like text, paving the way for new possibilities in communication, assistance, and creativity in programming.

Click to go back to exercise 3D

 # UNIT 4

Answers

Exercise 4A

1. sort out / reorganise / Can you rearrange the code? It's a bit messy
2. rule out / decide something isn’t an option / We can eliminate network issues as the cause of this bug
3. iron out / resolve issues / We still have a few problems that need to be resolved
4. take out / remove something / Try removing the function and running it again.
5. test out / check if something is working / Did you try the new feature?
6. figure out / find the answer to something / I'm trying to determine why the output is incorrect

Exercise 4B

1. The new feature should be implemented according to the specifications.
2. Was the team informed about the upcoming fixes to the codebase?
3. The devs need to be shown how to tackle those security issues we found.
4. Was this issue not (*wasn’t this issue*) detected before it went into production?
5. Why wasn't this bug caught during testing?

Exercise 4C

1. de-FAULTS
2. tran-SI-tion
3. im-PORT
4. con-FLICT
5. de-LE-tion
6. ex-TEN-sion
7. DE-fault
8. ex-CEP-tion

Exercise 4D

1. workaround
2. legacy issue
3. end-user
4. scope
5. run into
6. come across
7. good to go
8. straightforward
9. trigger
10. crashing
11. reproduce

 # listening

Daily Scrum

Transcript for Exercise 4D

1. Some end-users are experiencing difficulties with the login feature, particularly across the web and mobile platforms. I think this is because of a legacy issue within our authenticator. I'm going to implement a temporary workaround and just make sure people can still access the system while we figure out a permanent fix.
2. Given the scope of the affected users, and its impact, I'm going to suggest that we prioritize this in the upcoming sprint.
3. I was wondering if anyone else has run into similar issues? Joe, didn't you come across something like this recently?
4. Yeah, it was a fairly straightforward fix. I think it should be good to go now. So once the pull request is approved, I'll merge it into the main branch and then it can be deployed to staging for testing. Is anyone free to take a look this morning?
5. Yeah, it seems like the app is crashing everytime a user attempts to upload a large file, usually bigger than 50mb. I'm going to try and reproduce this bug - I'll find a large file to upload and then we can see if it triggers the same issue.

Click to go back to exercise 4D

 # UNIT 5

Answers

Exercise 5A

1. down the rabbit hole
2. up to speed
3. in the pipeline
4. put out fires
5. on the back burner

Exercise 5B

1. get up to speed
2 . on the back burner
3. in the pipeline

Exercise 5C

1. We identified key trends during the data analysis.
2. (Already in active voice)
3. Joe created an interactive dashboard for visualising the data.
4. We provided recommendations for optimisation based on the analysis.
5. Tools like Tableau provide valuable insights when analysing large datasets.
6. (Already in active voice)

Exercise 5D

1. A
2. B
3. B
4. A
5. B
6. A
7. B

Exercise 5E

1. No-meeting Friday's
2. To finish any uncompleted work
3. Moving on to
4. Not thorough enough (depth), Not enough coverage (range)
5. In terms of the...
6. How about,...
7. False! Early next week implies at the beginning of the week, i.e. Mon or Tues

 # listening

Retrospective Meeting

Transcript for Exercise 5E

Ok, let's start the retrospective, and we'll begin with what went well. So, I can see that the top card here was “no-meeting Friday's was a big hit with everyone” and I think we can all agree with that one. It was really nice to just concentrate on our work on Friday’s and not be distracted or interrupted by any meetings, and especially so we could tie up any loose ends before the weekend.

Moving on to what needs improving, ok so the comment 'too many in-sprint bugs being found in story acceptance testing', who wrote this one?

Yep! That was me. Bugs found that this stage are causing delays in completing the user stories, as a result we're missing some sprint deadlines. I'd say this is due to unclear requirements like the lack of detail in criteria of the user story, also perhaps the unit testing are not thorough enough.

Ok, let's think about how we could improve this. How about, increasing the frequency of the refinement sessions, and the depth that we go into, to make sure that the user stories are well defined?

Yeah, I'd like to allocate dedicated time to that if possible. In terms of the unit testing, there's not enough coverage, so some of the code is being missed.
How about we get the developers together and identify what's missing?

Sure, yep we can do that. I'll make a note of that now.
And I'll schedule something for early next week.

Click to go back to exercise 5E

Finished

Well done!

Want more?
Head to our website or instagram to check out what else we have for you:

www.speaktechenglish.com
@speaktech.english