ENGLISH FOR PROGRAMMERS

MASTER YOUR WORKFLOW IN ENGLISH:

REVIEW CODE - DISCUSS BUGS - AGILE PRACTICES

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

www.speaktechenglish.com

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

Unit 1. Implementing Code

vocabulary	action verbs	7
grammar	imperative present tense	9
pronunciation	keyboard symbols	11
listening	syntax	13

Unit 2. Code Review & Testing

vocabulary	noun phrases	15
grammar	parallel structure	17
pronunciation	connected speech	19
listening	code review	21

Unit 3. Discussing Code

vocabulary	modifiers	23
grammar	placement of modifiers	25
pronunciation	-ed verb	27
listening	chatGPT	29

Unit 4. Bug Fixing

vocabulary	phrasal verbs	31
grammar	passive voice	33
pronunciation	word stress	35
listening	daily scrum	37

Unit 5. Collaboration & Meetings

vocabulary	idioms	39
grammar	active voice	41
pronunciation	common mispronunciations	43
listening	retrospective meeting	45

ANSWERS & TRANSCRIPTS 46

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)

You have reached the end of the free sample

Get the full 59-page ebook with all 5 units and audio.

Get English for Programmers

Instant PDF download. Secure checkout with Stripe and PayPal.

speaktechenglish.com/product/english-for-programmers/