

# GitHub Copilot Certification

Session 2 – Exam Preparation Handout

Cluster Reply Copilot Campaign

February 2026

**Exam Overview:** 60 questions | 120 minutes | 70% passing score

**Focus Areas:** SKU Differences (31%) | Data Pipeline (15%) | Responsible AI (7%)

## Contents

<b>1 Responsible AI – The Six Principles</b>	<b>2</b>
1.1 Key Details per Principle . . . . .	2
<b>2 AI Limitations &amp; Risks</b>	<b>2</b>
<b>3 GitHub Copilot Plans (SKU Differences)</b>	<b>3</b>
<b>4 Data Pipeline – How Copilot Processes Code</b>	<b>3</b>
4.1 Data Retention Policies . . . . .	3
<b>5 Ways to Interact with Copilot</b>	<b>4</b>
5.1 Method 1: Inline Suggestions (Ghost Text) . . . . .	4
5.2 Method 2: Copilot Chat . . . . .	4
5.3 Method 3: Copilot CLI . . . . .	4
<b>6 Prompt Engineering</b>	<b>4</b>
6.1 Context Sources (Know All Five) . . . . .	4
6.2 Prompting Techniques . . . . .	4
<b>7 Key Exam Facts – Quick Reference</b>	<b>5</b>

## 1 Responsible AI – The Six Principles

Microsoft's Responsible AI framework governs all AI products including GitHub Copilot. **Memorize all six principles.**

#	Principle	Core Concept
1	<b>Fairness</b>	AI treats every user equitably, avoiding systematic bias
2	<b>Reliability &amp; Safety</b>	Systems perform predictably and minimize risks
3	<b>Privacy &amp; Security</b>	Protects code, data, and usage patterns
4	<b>Inclusiveness</b>	Benefits everyone regardless of ability or location
5	<b>Transparency</b>	Reveals how AI models make decisions
6	<b>Accountability</b>	Assigns clear ownership for AI outcomes

Table 1: The Six Pillars of Responsible AI

**💡 Exam Tip:** Memorization: “**FRPITA**” = Fairness, Reliability, Privacy, Inclusiveness, Transparency, Accountability. Expect 3–5 questions (7% of exam).

### 1.1 Key Details per Principle

**Fairness** Consistent, impartial suggestions regardless of background or language.

**Reliability** Rigorous testing against edge cases; resilient to malicious inputs.

**Privacy** Azure encryption, HSMs, Business/Enterprise code never trained on.

**Inclusive** Screen reader support, keyboard navigation, regional conventions.

**Transparent** `/explain` command, debugging tools, enterprise dashboards.

**Accountable** Define roles, monitor performance, conduct audits, address harms.

**⚠ Critical:** YOU own the code you commit – not Copilot, not GitHub, not Microsoft. You review it. You commit it. You are accountable.

## 2 AI Limitations & Risks

Understanding what Copilot **cannot** do is critical for the exam.

### ⚠ Limitations

- Training data may be outdated
- Limited context window
- Cannot perform calculations
- May suggest insecure patterns
- Bias from training data
- No true understanding of intent

### ✓ Mitigations

- Always review & validate output
- Use code scanning tools
- Apply security best practices
- Test thoroughly before deploy
- Understand context window
- Provide clear, specific prompts

**💡 Exam Tip:** “Copilot is an assistant, not a replacement – **YOU** are responsible for the code.”

### 3 GitHub Copilot Plans (SKU Differences)

This is the **MOST IMPORTANT** section – 31% weight means approximately 18–19 questions.

Feature	Free	Pro	Pro+	Business	Enterprise
Price/month	\$0	\$10	\$39	\$19/user	\$39/user
Completions	2,000/mo	Unlimited	Unlimited	Unlimited	Unlimited
Premium Requests	50/mo	300/mo	1,500/mo	300/user	1,000/user
IP Indemnity	✗	✗	✗	✓	✓
No Model Training	✗	✗	✗	✓	✓
Knowledge Bases	✗	✗	✗	✗	✓
Policy Management	✗	✗	✗	✓	✓
Coding Agent	✗	✓	✓	✓	✓
All AI Models	✗	✗	✓	✗	✗

Table 2: GitHub Copilot Plan Comparison (Overage: \$0.04/request)

💡 **Exam Tip:** Common exam traps: “Which plans include IP indemnity?” → **Business and Enterprise ONLY**. “Enterprise requires what?” → **GitHub Enterprise Cloud subscription**.

### 4 Data Pipeline – How Copilot Processes Code

Understanding the four-stage pipeline is essential – 15% of exam ( 9 questions).



Figure 1: Copilot Data Pipeline

#### Stage 1: IDE Context

- Current file content & cursor
- Open neighboring tabs
- Repository structure, file paths
- Chat history

#### Stage 2: Proxy Server

- Authentication/authorization
- PII detection
- Toxic language screening
- Jailbreak prevention, rate limiting

#### Stage 3: LLM

- Hosted in GitHub-owned Azure tenants
- Processes prompt with context
- Multiple models per plan

#### Stage 4: Post-Processing

- Public code filter (150+ chars)
- Quality verification
- Final formatting

#### 4.1 Data Retention Policies

Data Type	Retention
User engagement data	2 years
Chat history	28 days
Business/Enterprise prompts	Never used for training
Individual plan prompts	May contribute (opt-out available)

💡 **Exam Tip: Public Code Filter:** Blocks suggestions matching public code **>150 characters**. 1% of suggestions match. Can be enabled/disabled per org policy.

## 5 Ways to Interact with Copilot

### 5.1 Method 1: Inline Suggestions (Ghost Text)

Action	Windows	Mac
Accept suggestion	Tab	Tab
Dismiss suggestion	Esc	Esc
Next alternative	Alt+]	Option+]
Previous alternative	Alt+[	Option+[

Table 3: Inline Suggestion Shortcuts

### 5.2 Method 2: Copilot Chat

Action	Windows	Mac
Open Chat Panel	Ctrl+Alt+I	Cmd+Option+I
Inline Chat	Ctrl+I	Cmd+I

Table 4: Chat Shortcuts

**Slash Commands (MEMORIZE):** /explain (explain code) | /fix (fix bugs) | /tests (generate tests) | /doc (documentation) | /optimize (performance)

### 5.3 Method 3: Copilot CLI

- `gh copilot explain` – Explain a command or concept
- `gh copilot suggest` – Get command suggestions

## 6 Prompt Engineering

### 6.1 Context Sources (Know All Five)

1. **Current File** – Primary context (content before/after cursor)
2. **Open Tabs** – Related files you have open
3. **File Paths** – Directory names provide semantic hints
4. **Comments** – Natural language descriptions guide suggestions
5. **Chat History** – Previous exchanges inform future responses

### 6.2 Prompting Techniques

Technique	Description	When to Use
<b>Zero-Shot</b>	No examples given	Simple and common programming tasks
<b>One-Shot</b>	Single example given	Establishing a specific output format
<b>Few-Shot</b>	Multiple examples (2–5)	Complex patterns where consistency is critical
<b>Chain-of-Thought</b>	Step-by-step reasoning	Complex reasoning and debugging tasks

Table 5: Prompting Techniques

💡 **Exam Tip: Chain-of-Thought magic phrases:** “Let’s think step by step” | “Think through this carefully” | “Break this down” | “Explain your reasoning”

**Prompt Quality:**

✗ Bad: "make a function"

✓ Good: "Create TypeScript function that validates emails, returns boolean"

## 7 Key Exam Facts – Quick Reference

1. ✓ **IP Indemnity** – Business and Enterprise ONLY
2. ✓ **Data Training** – Business/Enterprise code NEVER used for training
3. ✓ **Public Code Filter** – Blocks >150 character matches ( 1% of suggestions)
4. ✓ **Knowledge Bases** – Enterprise-only feature
5. ✓ **Premium Requests:** Free=50 | Pro=300 | Pro+=1,500 | Biz=300 | Ent=1,000
6. ✓ **User Responsibility** – Developer is accountable for AI-generated code
7. ✓ **Context Sources** – File, tabs, paths, comments, chat history (5 sources)
8. ✓ **6 AI Principles** – Fairness, Reliability, Privacy, Inclusiveness, Transparency, Accountability

### 🎓 Exam Strategy

**Time:** 2 min/question average | **Flag & Return:** Mark difficult questions  
**Focus:** SKU Differences (31%), Data Pipeline (15%), Responsible AI (7%)

**Resources:** Microsoft Learn | GitHub Docs | GitHub Skills

*Good luck with your certification!*

Cluster Reply GmbH | <https://www.reply.com/cluster-reply-germany>