



Certified Mobile Application Engineer (CMAE)

Ten Sample Questions

These ten questions are written at the difficulty of the live exam. Six carry a figure, a data table, a configuration or a code listing; four are plain scenario questions, as on the live paper. Every one is a scenario a working mobile application engineer would meet, and none of them appears in the live question bank.

The live exam is multistage adaptive: how you score on one stage selects the difficulty of the next, so no two candidates sit the same paper. It is a performance-based exam: each form can carry up to 5 to 12 performance-based questions alongside the multiple-choice questions, which means you build, sequence or complete something rather than only choosing from a list. Answers and full explanations follow each question here; the live exam does not disclose item-level answers.

Question 1

Domain 1: Mobile Foundations

DATA

Sarah Miller reviews the app lifecycle callbacks below.

Callback	Purpose
On start	Becomes visible
On resume	Gains focus
On pause	Losing focus
On stop	No longer visible

Table 1.1 Lifecycle callbacks and purpose.

When should the app release the camera and sensors?

- ☐ A Free them on pause
- ☐ B Free them on redraw
- ☐ C Keep them always open
- ☐ D Free only on force quit

Correct answer: A

A mobile app should release scarce hardware such as the camera and sensors when it loses focus and is paused, so a backgrounded app does not hold resources other apps need. Freeing on every redraw is wasteful, keeping them open drains the battery, and force quit is not a reliable moment.

Question 2

Domain 1: Mobile Foundations

CONFIG

James Carter reviews a build config that reaches very few devices.

```
minSdk = newest only
target = older release
result = few devices
installs = falling
```

Listing 2.1 The current build config.

What is the soundest change?

- ☐ A Raise the minSdk higher
- ☐ B Lower minSdk sensibly
- ☐ C Drop older devices
- ☐ D Target an older API

Correct answer: B

Setting the minimum SDK to the newest release only excludes most devices in the field, so lowering it to a sensible floor widens reach while keeping a modern target. Raising it further shrinks reach, dropping devices abandons users, and lowering the target API loses newer platform behaviour.

Question 3

Domain 2: UI and UX

DATA

Olivia Davis reviews layout values that are all set in fixed pixels.

Property	Current value
Padding	16 fixed px
Font size	14 fixed px
Icon	24 fixed px
Margin	8 fixed px

Table 3.1 Sizes set in fixed pixels.

What keeps the layout consistent across screen densities?

- ☐ A Use fixed pixels
- ☐ B Use scalable units
- ☐ C One layout per phone

- D** Hard-code all sizes

Correct answer: B

Fixed pixels render at different physical sizes across densities, so density-independent, scalable units keep spacing and text consistent on every screen. A layout per phone does not scale, and hard-coded sizes repeat the original fault.

Question 4

Domain 3: Data and Networking

CONFIG

Emma Wilson reviews a data path that freezes the interface.

```
load() on main thread
no cache layer
UI freezes on load
retry = none
```

Listing 4.1 The current data path.

What is the soundest change?

- A** Fetch on the main thread
B Block UI until done
C Remove all networking
D Fetch off main, cache

Correct answer: D

Network work on the main thread freezes the interface, so it belongs on a background thread with a cache to serve repeat reads and a retry for transient failures. Fetching on the main thread and blocking the interface repeat the fault, and removing networking defeats the feature.

Question 5

Domain 4: Device Capabilities and Integration

DATA

Liam Anderson reviews which capabilities require a permission.

Capability	Needs permission
Show a map	No
Get location	Yes
Read contacts	Yes
Play a sound	No

Table 5.1 Capabilities and permission needs.

When should the app request the location permission?

- A** Ask for all at install
B In context, when used
C Never ask the user

- ☐ D Demand it upfront

Correct answer: B

Requesting a sensitive permission such as location in context, at the moment the feature needs it, raises the grant rate and respects the user. Asking for everything at install or upfront feels intrusive, and never asking breaks the feature.

Question 6

Domain 5: Testing, Performance and Quality

DATA

Ava Thompson reviews how quality is checked on the team today.

Observation	Result
Automated tests	None
Manual demo	Passed
Tested on many devices	No
Re-run after change	No

Table 6.1 How quality is checked today.

What best improves reliability here?

- ☐ A Automated test suite
- ☐ B Rely on manual demos
- ☐ C Ship without testing
- ☐ D Test on one device only

Correct answer: A

A single manual demo proves little, so an automated test suite run on each change catches regressions across devices and configurations. Relying on demos judges quality by anecdote, shipping untested defers faults to users, and one device hides fragmentation issues.

Question 7

Domain 3: Data and Networking

An app becomes unusable the moment the network drops. What is the soundest design?

- ☐ A Always assume a network
- ☐ B Show a blank screen
- ☐ C Cache and queue writes
- ☐ D Block the whole app

Correct answer: C

A mobile app should expect intermittent connectivity, so caching reads and queuing writes lets it stay useful offline and sync when the network returns. Assuming a network, showing a blank screen, or blocking the app all fail the user when connectivity drops.

Question 8

Domain 6: Security, Release and Operations

An API key is hard-coded in the shipped app binary, where anyone can extract it. What is required?

- ☐ A Leave key in the code
- ☐ B Keep keys off device
- ☐ C Log the key to console
- ☐ D Share key in the URL

Correct answer: B

A secret baked into the binary can be extracted from the package, so keys belong on a backend the app calls, with any device token held in secure platform storage and rotated. Leaving it in code, logging it, or placing it in a URL all expose the secret.

Question 9

Domain 4: Device Capabilities and Integration

The app polls GPS every second in the background and drains the battery. What is the soundest fix?

- ☐ A Poll GPS more often now
- ☐ B Never use location
- ☐ C Keep GPS on always
- ☐ D Batch and lower rate

Correct answer: D

Continuous high-rate location in the background is a major battery cost, so lowering the update rate, batching updates, and using coarser accuracy when it suffices cuts drain while meeting the need. Polling faster or keeping GPS on always worsens it, and never using location breaks the feature.

Question 10

Domain 5: Testing, Performance and Quality

Scrolling stutters because large images are decoded on the interface thread. What best fixes this?

- ☐ A Decode off UI thread
- ☐ B Decode on UI thread
- ☐ C Load full-size images
- ☐ D Disable all scrolling

Correct answer: A

Decoding large images on the interface thread blocks each frame and causes visible stutter, so decoding and resizing off the main thread keeps scrolling smooth. Decoding on the interface thread repeats the fault, full-size images waste memory, and disabling scrolling removes the feature.

