



OOPS ABAP-15 questions

Q1 What is encapsulation in ABAP OO?

Hiding internal data using private/protected visibility. External code accesses only via public methods. Prevents accidental modification.

✓ Give a real class example-ZCL_CREDIT_CHECK

Q2 Interface vs abstract class?

Interface = contract only, no implementation. Abstract class = partial implementation, forces subclasses to complete it. Use interface when multiple unrelated classes share a contract.

Q3 What is polymorphism in ABAP?

Same method name, different behaviour per class. Achieved via interface references or inheritance. Allows generic code to handle different object types.

Q4 Static vs instance method?

Static (CLASS-METHODS) = called on class directly, no object needed. Instance method = called on object. Use static for utility functions with no state.

Q5 What is a constructor in ABAP?

Method CONSTRUCTOR runs automatically on object creation. Used to initialise attributes. Cannot be called explicitly-only triggered by CREATE OBJECT.

Q6 What is inheritance and when to use it?

Subclass inherits methods/attributes of superclass via INHERITING FROM. Use when classes share behaviour but differ in specifics. Prefer composition over deep inheritance chains.

Q7 What is method redefinition?

Subclass overrides a superclass method using REDEFINITION keyword. Allows subclass-specific behaviour. Call SUPER->METHOD to retain parent logic if needed.

✓ Explain the SUPER-> call-shows depth

Q8 What is a singleton pattern in ABAP?

Ensures only one instance of a class exists. Achieved by private constructor + static GET_INSTANCE method. Used for shared resources like configuration objects.

Q10 What are events in ABAP OO?

Publisher-subscriber pattern. The class raises an EVENT; handler classes register via SET HANDLER. Decouples sender from receiver-useful in ALV and UI frameworks.

Q11 What is a factory method pattern?

Static method that returns an object instance based on input parameters. Hides object creation logic. Common in SAP standard classes like CL_SALV_TABLE=>FACTORY.

Q12 Difference between TYPE REF TO class and interface?

TYPE REF TO class = concrete reference, access class-specific methods. TYPE REF TO interface = generic reference, polymorphic-preferred for loose coupling.

Q13 What is exception class handling in ABAP?

Exceptions are classes inheriting from CX_ROOT. Raise with RAISE EXCEPTION TYPE. Catch with TRY/CATCH/CLEANUP/ENDTRY. Preferred over SY-SUBRC for complex error handling.

Q14 What is a friendship relationship?

One class grants another access to its private/protected components using FRIENDS addition. Used sparingly-breaks encapsulation. Common in test frameworks.

Q15 When to use OO vs procedural ABAP?

OO = complex logic, reusable components, testable code, S/4HANA and RAP standard. Procedural = simple reports, legacy maintenance. Modern ABAP is always OO.

✓ Say "All new development I write is OO"

BDC & data migration — 10 questions

Q1 Session method vs call transaction?

aSession method = asynchronous, better error log, recommended for large volumes. Call Transaction = synchronous, immediate processing, small volumes or online use.

Q2 How does BDC recording work?

Transaction SHDB records screen field names and values. Output = BDC table with BDCDATA structure. Replay fills screens programmatically without user interaction.

Q3 When would you use LSMW over BDC?

LSMW when no coding skills needed, one-time migration, standard business objects available. BDC when complex logic, conditions, or multiple screens needed in one load.

Q4 How do you handle BDC errors?

Collect records with MODE 'E' (error display). Log failed records with their BDC messages. Create error report with original data + error reason for reprocessing.

✓Mention error log table and reprocessing strategy

Q5 What is a direct input method?

SAP function modules that load data directly into DB tables, bypassing screen processing. Fastest method. Examples: MATERIAL_MAINTAIN_ITEM, VENDOR_CREATE_FROM_DATA.

Q6 What is BDCDATA structure?

Internal table structure for BDC. Key fields: PROGRAM, DYNPRO (screen number), DYNBEGIN (Y = new screen), FNAME (field name), FVAL (value).

Q7 How do you handle multiple sessions in BDC?

Use KEEP_SCROLL_POSITION and split data into batches. Each session processes a defined number of records. Monitor via SM35-session management transaction.

Q8 What is data migration cutover strategy?

Mock run → reconcile → sign-off → freeze legacy → final load → validate → go-live. Delta load handles transactions between mock and final cutover.

Q9 BAPI vs BDC for data migration?

BAPI preferred: faster, no screen dependency, better error handling, officially supported. BDC is used when no BAPI exists. BAPI errors in RETURN table, not screen messages.

Q10 What is MIGRATION COCKPIT in S/4HANA?

SAP's modern migration tool-replaces LSMW for S/4HANA. Template-based, supports staging tables and direct transfer. Preferred for clean-core compliant migrations.

✓List these transactions-proves hands-on experience

IDoc & RFC-10 questions

Q1 What is an IDoc?

Intermediate Document-SAP's standard data container for async message exchange between systems. Has control record (EDI_DC40), data segments, status records.

Q2 IDoc vs BAPI vs RFC-when to use each?

IDoc = async, message-based, guaranteed delivery, EDI. BAPI = sync, business objects, external systems. RFC = real-time function call between SAP systems.

Q3 IDoc status codes to know?

03 = sent to application. 51 = error in application. 53 = posted. 64 = IDoc ready to transfer. Status 51 is what you will debug most in production support.

✓ Knowing status 51 shows real production experience

Q4 What is a message type and basic type?

Message type = business purpose (ORDERS, INVOIC, MATMAS). Basic type = IDoc structure definition (segments and fields). One message type can use multiple basic types.

Q5 How do you extend an IDoc?

Create extension type with custom Z-segments. Assign to basic type via IDOCTYP. Use user exits WE30 (add segment), WE20 (partner profile), IDOC_INPUT_* for inbound enhancement.

Q6 What is RFC and its types?

Remote Function Call-calls FM on another system. Types: sRFC (sync, waits for response), aRFC (async, no wait), tRFC (transactional, exactly-once), qRFC (queued order).

Q7 tRFC vs qRFC difference?

tRFC = transactional, exactly-once execution, no order guarantee. qRFC = queued RFC, processes in a defined sequence-used when order execution matters between calls.

Q8 How do you reprocess a failed IDoc?

Transaction BD87 for inbound, WE19 for test/reprocess. Fix root cause first. Reprocess via program RBDMANI2 or manually from WE02/WE05. Document the fix in transport.

Q9 What is partner profile in IDoc?

Transaction WE20-defines inbound/outbound IDoc processing rules per partner. Specifies message type, process code, port, and logical message. Must be configured before IDocs flow.

Q10 IDoc monitoring transactions to know?

WE02 = IDoc list. WE05 = IDoc display. BD87 = inbound processing status. WE19 = test tool. Knowing these transactions shows real integration project experience.

✓List these transactions-proves hands-on experience

"How to explain your project" - interview template

Context:

"I worked on [project type] for a [industry] client on [ECC/S4HANA]."

Your role:

"I was responsible for [your specific area - OOPS classes / IDoc mapping / BDC programs]."

Challenge:

"The main challenge was [specific technical probleme.g. performance on 500k records]."

Solution:

"I solved it by [your approach-e.g. replacing SELECT loop with JOIN and FOR ALL ENTRIES]."

Result:

"The outcome was [runtime reduced by X%/migration completed in Y hours/zero errors in UAT]."

Salary negotiation tip-mid-level

- **Research market rate before interview.** Sites: LinkedIn Salary, Glassdoor, IT Jobs Watch (UK), Levels.fyi (US/global).

- **When asked:** "Based on my 3 years in OOPS ABAP and IDoc integration, I am targeting [X-Y range]. "Always give a range, not a single number.

- **Anchor high-say the top of your range first.** Negotiation goes down, never up from your first number.

- **If they push back:** "I am open to discussing the full package-including growth path and review timeline."

- **Never reveal your current salary.** Redirect: "I am focused on the market rate for this role and my skills."

**Laeq Siddique**

Free resource by Cremencing Solutions cremencing.com
CS Peach-Alconsultant for SAP ABAP teams



CREMENCING
SOLUTIONS

Project template — 1 line each

- **Context:** client + system + industry
- **Role:** your specific area
- **Challenge:** the real technical problem
- **Solution:** what you did
- **Result:** measurable outcome

Salary — 3 rules

- **Give a range — never a single number**
- **Anchor high — say top of range first**
- **Never reveal current salary**
- **Redirect: "I am focused on market rate"**
- **Push back: "Open to full package discussion"**

Bonus tips for blogs

- **Post this card on LinkedIn as a carousel**
- **Use OOPS section as blog post #1 teaser**
- **Use IDoc section for blog post #3 teaser**
- **mail this to list day before interview season**
- **Pin to Cremencing Instagram as study story**

OOPS ABAP — top 5
Most asked at mid-level**Encapsulation**

Hide data with private visibility. Expose only via public methods. Say: I always set attributes private.

Interface vs abstract class

Interface = contract only. Abstract = partial impl. Use interface for loose coupling.

Singleton pattern

Private constructor + static GET_INSTANCE. One instance only. Used for config objects.

Exception class handling

TRY/CATCH with CX_ROOT subclasses. Preferred over SY-SUBRC for complex flows.

When to use OO vs procedural

All new dev = OO. Procedural only for legacy maintenance. S/4HANA is always OO.

BDC & migration — top 5
Data load techniques**Session vs call transaction**

Session = async, large volumes. Call Transaction = sync, small volumes.

BAPI vs BDC for migration

BAPI preferred — faster, no screen dependency, better errors, officially supported.

Error handling in BDC

Collect errors with MODE 'E'. Log failed records. Build reprocessing report.

LSMW vs Migration Cockpit

Migration Cockpit = S/4HANA standard. LSMW = legacy. Always mention Cockpit for S/4.

Cutover strategy

Mock run → reconcile → freeze legacy → final load → validate → go-live + delta.

IDoc & RFC — top 5
Integration must-knows**IDoc vs BAPI vs RFC**

IDoc = async, EDI. BAPI = sync, business obj. RFC = real-time remote call.

IDoc status 51

Error in application. Most common production issue. Fix app error, reprocess via BD87.

RFC types

sRFC (sync) · aRFC (async) · tRFC (exactly-once) · qRFC (queued order).

Reprocess failed IDoc

Fix root cause first. Then BD87 for inbound. WE19 for test. RBDMANI2 for bulk.

Monitoring transactions

WE02 list · WE05 display · BD87 status · WE19 test. Know all four.

**Laeeq Siddique**
 Free resource by Cremencing Solutions cremencing.com
 CSPeach-Alconsultant for SAP ABAP teams