📘 MCQs: Pydantic dataclasses as output\_type in Agents

### 1. What is the purpose of setting a dataclass as an agent’s output\_type?

A) To reduce memory usage  
B) To enforce structured, validated responses  
C) To generate random outputs  
D) To disable schema validation

**Answer:** B  
**Explanation:** Using dataclass as output\_type ensures agents always return structured data following a defined schema.

### 2. Which Python decorator is used to define a Pydantic dataclass?

A) @model  
B) @dataclass from pydantic.dataclasses  
C) @BaseModel  
D) @schema

**Answer:** B  
**Explanation:** @pydantic.dataclasses.dataclass is used to define validated dataclasses.

### 3. Why is a dataclass better than a plain dictionary for output\_type?

A) It is faster  
B) It enforces type safety and validation  
C) It removes the need for serialization  
D) It automatically optimizes GPU usage

**Answer:** B  
**Explanation:** Unlike dictionaries, dataclasses enforce **types and validation**, reducing errors in AI-agent responses.

### 4. In Agentic AI, why would we use a dataclass instead of BaseModel for outputs?

A) Dataclass is simpler and lightweight  
B) BaseModel does not support serialization  
C) Dataclass ignores types  
D) BaseModel is deprecated

**Answer:** A  
**Explanation:** Dataclasses are more **lightweight** when full Pydantic features (like .json()) are not required.

### 5. If an agent is expected to return a ****name and age****, how would you define the dataclass?

A)

@dataclass

class Person:

name: str

age: int

B)

class Person(BaseModel):

name: int

age: str

C) Dict[str, Any]  
D) A tuple

**Answer:** A  
**Explanation:** A simple @dataclass with type hints enforces correct schema for outputs.

### 6. What happens if the AI agent outputs a wrong type (e.g., "25" for int age)?

A) It is ignored  
B) Pydantic attempts type coercion  
C) It always crashes  
D) It stores as string

**Answer:** B  
**Explanation:** Pydantic dataclasses attempt type coercion before raising errors.

### 7. Which is TRUE about dataclass used as output\_type?

A) It disables validation  
B) It enforces schema but is lighter than BaseModel  
C) It is slower than BaseModel  
D) It cannot be used with agents

**Answer:** B  
**Explanation:** Pydantic dataclasses strike a balance: **schema enforcement + lightweight design**.

### 8. Why are dataclasses useful in multi-agent systems?

A) They allow agents to communicate using structured objects  
B) They reduce API costs  
C) They remove need for prompts  
D) They disable memory

**Answer:** A  
**Explanation:** Dataclasses ensure structured, predictable outputs across different agents.

### 9. Which method is commonly used to convert a dataclass into a dictionary?

A) .dict()  
B) asdict() from dataclasses  
C) .to\_json()  
D) .schema()

**Answer:** B  
**Explanation:** Standard dataclasses use asdict() for serialization.

### 10. Can nested dataclasses be used as agent output\_type?

A) Yes, Pydantic validates nested dataclasses too  
B) No, only flat dataclasses are allowed  
C) Only with JSON  
D) Not supported

**Answer:** A  
**Explanation:** Pydantic supports **nested dataclasses**, enabling complex structured outputs.

### 11. What is the main limitation of dataclass compared to BaseModel?

A) No support for nested models  
B) No built-in .json() or .dict() methods  
C) No type validation  
D) Cannot be used in agents

**Answer:** B  
**Explanation:** Unlike BaseModel, dataclasses don’t include .json() or .dict() helpers by default.

### 12. Which agent use case benefits MOST from dataclass outputs?

A) Free-text storytelling  
B) Returning structured results like API responses or predictions  
C) Image rendering  
D) Token sampling

**Answer:** B  
**Explanation:** Dataclasses work best when an agent must **return structured results**.

### 13. In Pydantic dataclasses, what enforces type checking?

A) Python runtime  
B) Pydantic validation layer  
C) JSON Schema  
D) Nothing, they are unchecked

**Answer:** B  
**Explanation:** Pydantic validates inputs at runtime, unlike plain dataclasses.

### 14. Which code snippet shows correct use of dataclass as agent output?

A)

@dataclass

class Result:

success: bool

B)

class Result:

success: bool

C)

class Result(BaseModel):

success: bool

D)

Result = dict(success=bool)

**Answer:** A  
**Explanation:** Correct usage requires @dataclass with type annotations.

### 15. Why might you choose dataclasses over dictionaries for outputs?

A) Dictionaries cannot store nested data  
B) Dataclasses improve readability, validation, and maintainability  
C) Dataclasses are faster in all cases  
D) Dataclasses avoid serialization

**Answer:** B  
**Explanation:** Dataclasses provide **structured clarity**, unlike untyped dictionaries.

### 16. Which feature is unique to Pydantic dataclasses compared to standard Python dataclasses?

A) Type annotations  
B) Runtime validation  
C) Field defaults  
D) Nested structures

**Answer:** B  
**Explanation:** Pydantic adds **validation**, unlike plain Python dataclasses.

### 17. If you want ****minimal overhead**** in an agent output schema, which should you pick?

A) BaseModel  
B) dataclass  
C) Plain dict  
D) NamedTuple

**Answer:** B  
**Explanation:** Dataclasses are more lightweight than BaseModel while still providing validation.

### 18. What happens if an agent’s output does not match the dataclass schema?

A) It silently fixes  
B) A ValidationError is raised  
C) It always converts to string  
D) It is skipped

**Answer:** B  
**Explanation:** Schema mismatches trigger **Pydantic validation errors**.

### 19. In Fundamentals of Agentic AI, structured outputs (via dataclasses) mainly solve:

A) Token sampling issues  
B) Ambiguity in agent communication  
C) Prompt length limits  
D) GPU memory issues

**Answer:** B  
**Explanation:** Structured outputs reduce **ambiguity** when agents exchange data.

### 20. Best practice: When should you use dataclasses as output\_type in Agentic AI?

A) For unstructured free text  
B) For structured, schema-bound results with low overhead  
C) Always, even for text  
D) Never, use BaseModel instead

**Answer:** B  
**Explanation:** Dataclasses are ideal when you need **structured outputs** but don’t need the full power of BaseModel.