4: Object-oriented programming

Answer these questions

Question 1

Classes have a special method that serves as a blueprint for creating objects of that class. What is the method called?

□ A builder

□ An instantiator

✅ A constructor

□ A blueprint

Question 2

Which of the following statements about interfaces and abstract classes is NOT correct?

□ Abstract classes can have constructors.

□ Interfaces can't have constructors.

✅ Interfaces and abstract classes can be instantiated directly.

□ Abstract properties must be implemented by subclasses of the abstract class.

Question 3

Which of the following is NOT a Kotlin visibility modifier for properties, methods, etc.?

□ `internal`

✅ `nosubclass`

□ `protected`

□ `private`

Question 4

Consider this data class: `data class Fish(val name: String, val species:String, val colors:String)` Which of the following is NOT valid code to create and destructure a `Fish` object?

□ `val (name1, species1, colors1) = Fish("Pat", "Plecostomus", "gold")`

□ `val (name2, _, colors2) = Fish("Bitey", "shark", "gray")`

□ `val (name3, species3, _) = Fish("Amy", "angelfish", "blue and black stripes")`

✅ `val (name4, species4, colors4) = Fish("Harry", "halibut")`

Question 5

Let's say you own a zoo with lots of animals that all need to be taken care of. Which of the following would NOT be part of implementing caretaking?

✅ An `interface` for different types of foods animals eat.

□ An `abstract Caretaker` class from which you can create different types of caretakers.

□ An `interface` for giving clean water to an animal.

□ A `data` class for an entry in a feeding schedule.