Singleton Pattern

1. **Which of the following statements about the Singleton pattern is true?**

   - A) Singleton allows multiple instances.

   - B) Singleton ensures only one instance of a class exists.

   - C) Singleton does not provide a global access point.

   - D) Singleton pattern is not thread-safe by default.

   - **Answer: B**

2. **What is the main purpose of the `lock` keyword in a Singleton implementation?**

   - A) To ensure that only one thread can access a resource at a time.

   - B) To prevent any thread from accessing a resource.

   - C) To allow multiple threads to create instances.

   - D) To make the Singleton instance immutable.

   - **Answer: A**

3. **Which of the following is a common problem with a basic Singleton implementation in a multithreaded environment?**

   - A) Multiple threads can create multiple instances.

   - B) The instance cannot be accessed globally.

   - C) The Singleton instance is mutable.

   - D) The instance cannot be created.

   - **Answer: A**

4. **In C#, which keyword is often used in Singleton to prevent creating instances using the `new` keyword?**

   - A) `private`

   - B) `protected`

   - C) `internal`

   - D) `public`

   - **Answer: A**

5. **Which of the following is NOT a typical characteristic of a Singleton class?**

   - A) A private constructor.

   - B) A public static method to access the instance.

   - C) Global visibility.

   - D) Public constructor.

   - **Answer: D**

Factory Method Pattern

6. **What does the Factory Method pattern achieve?**

   - A) It creates objects without specifying the exact class of object that will be created.

   - B) It limits object creation to one instance.

   - C) It allows cloning of objects.

   - D) It enforces a specific construction sequence.

   - **Answer: A**

7. **Which of the following is true about the Factory Method?**

   - A) It forces subclasses to create objects.

   - B) It uses reflection to create objects.

   - C) It defines an interface for creating an object but lets subclasses decide which class to instantiate.

   - D) It can only create a single type of object.

   - **Answer: C**

8. **Which of these is a primary benefit of using the Factory Method pattern?**

   - A) Ensures only one instance of a class.

   - B) Allows loose coupling by delegating object creation to subclasses.

   - C) Ensures thread safety in object creation.

   - D) Provides a global point of access to objects.

   - **Answer: B**

9. **Which of the following is typically part of the Factory Method pattern?**

   - A) Abstract Factory

   - B) Concrete Creator

   - C) Prototype

   - D) Singleton

   - **Answer: B**

10. **The Factory Method pattern is used in which of the following scenarios?**

    - A) When you need to create an object only once.

    - B) When a class cannot anticipate the class of objects it must create.

    - C) When you need to clone objects.

    - D) When the class to be instantiated is known at compile-time.

    - **Answer: B**

Abstract Factory Pattern

11. **What is the main goal of the Abstract Factory pattern?**

    - A) To create a single object.

    - B) To create families of related or dependent objects.

    - C) To create objects that are unrelated.

    - D) To implement object cloning.

    - **Answer: B**

12. **Which of the following is true about the Abstract Factory pattern?**

    - A) It provides an interface to create a single object.

    - B) It is used to create clones of objects.

    - C) It provides an interface for creating families of related or dependent objects without specifying their concrete classes.

    - D) It allows only one concrete implementation.

    - **Answer: C**

13. **The Abstract Factory pattern is useful when:**
    - A) There is only one product to be created.
    - B) There are multiple families of related products to be created.
    - C) All objects are to be created in a single method.
    - D) You need to enforce thread safety.
    - **Answer: B**

14. **In the Abstract Factory pattern, the ConcreteFactory classes:**
    - A) Implement the AbstractFactory interface.
    - B) Inherit from the Product interface.
    - C) Are singleton classes.
    - D) Provide global access to objects.
    - **Answer: A**

15. **Which of the following is NOT a component of the Abstract Factory pattern?**
    - A) AbstractProduct
    - B) ConcreteProduct
    - C) ConcretePrototype
    - D) AbstractFactory
    - **Answer: C**

Builder Pattern

16. **What problem does the Builder pattern solve?**
    - A) It creates a single object.
    - B) It separates the construction of a complex object from its representation.
    - C) It allows for multiple instances of a class.
    - D) It ensures that an object can only be created once.
    - **Answer: B**

17. **Which of the following best describes the Builder pattern?**

- A) It provides a global access point to objects.
- B) It encapsulates the creation of a family of related objects.
- C) It constructs complex objects step by step.
- D) It clones objects using a prototype instance.
- **Answer: C**

18. **The Director in the Builder pattern is responsible for:**
    - A) Creating the product directly.
    - B) Defining the sequence of steps to construct a product.
    - C) Implementing the Product interface.
    - D) Cloning the product.
    - **Answer: B**

19. **Which of the following is an advantage of using the Builder pattern?**
    - A) It enforces the use of a single constructor.
    - B) It allows the creation of complex objects with multiple configurations.
    - C) It restricts object creation to one instance.
    - D) It avoids the need for subclasses.
    - **Answer: B**

20. **In the Builder pattern, what does the `ConcreteBuilder` class do?**
    - A) It defines an abstract interface for creating parts of a Product object.
    - B) It constructs and assembles parts of the product.
    - C) It acts as the Director of the pattern.
    - D) It provides global access to the product.
    - **Answer: B**

Prototype Pattern

21. **What is the Prototype pattern primarily used for?**
    - A) Creating a single instance of a class.

- B) Creating objects by copying an existing object.
- C) Creating a family of related objects.
- D) Defining the construction process of an object.
- **Answer: B**

22. **Which method is typically implemented in a class to support the Prototype pattern?**
- A) `Clone`
- B) `FactoryMethod`
- C) `Build`
- D) `CreateInstance`
- **Answer: A**

23. **The `MemberwiseClone` method in C# is used in the Prototype pattern to:**
- A) Create a shallow copy of the object.
- B) Create a deep copy of the object.
- C) Create a single instance of the object.
- D) Enforce thread safety during cloning.
- **Answer: A**

24. **Which of the following is true about the Prototype pattern?**
- A) It involves creating objects by calling a constructor.
- B) It allows the creation of new objects by copying an existing object.
- C) It provides a global point of access to the cloned object.
- D) It restricts the number of objects to be cloned.
- **Answer: B**

25. **When is it advantageous to use the Prototype pattern?**
- A) When creating new objects from scratch is costly.
- B) When only one instance of an object is needed.
- C) When objects need to be created with complex configurations.
- D) When the construction process is simple and does not vary.

- **Answer: A**

Mixed Questions

26. **Which pattern would you use to create a family of related objects?**
    - A) Singleton
    - B) Factory Method
    - C) Abstract Factory
    - D) Builder
    - **Answer: C**

27. **Which pattern allows you to create objects step by step?**
    - A) Singleton
    - B) Factory Method
    - C) Abstract Factory
    - D) Builder
    - **Answer: D**

28. **Which pattern provides a way to create objects without specifying the exact class of object that will be created?**
    - A) Singleton
    - B) Factory Method
    - C) Abstract Factory
    - D) Builder
    - **Answer: B**

29. **Which of the following patterns is best suited for ensuring that a class has only one instance?**
    - A) Singleton
    - B) Factory Method
    - C) Abstract Factory
    - D) Builder

- **Answer: A**

30. **Which pattern is useful when creating an object is resource-intensive and multiple copies of the object are required?**

- A) Singleton
- B) Prototype
- C) Factory Method
- D) Builder
- **Answer: B**

Singleton Advanced

31. **Which of the following statements about the Singleton pattern is false?**

- A) The Singleton pattern can be lazy-initialized.
- B) The Singleton pattern is always thread-safe.
- C) The Singleton instance is usually stored in a static field.
- D) Singleton can be combined with other patterns like Factory.
- **Answer: B**

32. **How can you prevent a Singleton class from being cloned?**

- A) By overriding the `Clone` method and throwing an exception.
- B) By using a `private` constructor.
- C) By marking the class as `sealed`.
- D) By using the `lock` keyword.
- **Answer: A**

33. **What is the primary disadvantage of the Singleton pattern?**

- A) It is not thread-safe.
- B) It is difficult to test because of its global state.

- C) It allows multiple instances.
- D) It cannot be lazy-initialized.
- **Answer: B**

34. **Which of the following scenarios is a bad use case for the Singleton pattern?**
    - A) Managing a database connection.
    - B) A global logging instance.
    - C) A configuration settings class.
    - D) A factory class creating different objects.
    - **Answer: D**

35. **How do you make a Singleton class thread-safe in C#?**
    - A) Use the `static` keyword.
    - B) Use a `lock` on the instance creation.
    - C) Use the `sealed` keyword.
    - D) Make the constructor `private`.
    - **Answer: B**

Factory Method Advanced

36. **In the Factory Method pattern, the `Creator` class:**
    - A) Creates the concrete product directly.
    - B) Defines the Factory Method to delegate creation to subclasses.
    - C) Implements the `IProduct` interface.
    - D) Creates a single instance of the product.
    - **Answer: B**

37. **Which pattern can be seen as a more general form of the Factory Method pattern?**
    - A) Singleton
    - B) Prototype
    - C) Abstract Factory

- D) Builder
- **Answer: C**

38. **What is a common use case for the Factory Method pattern?**
    - A) When a class requires a single instance.
    - B) When a class cannot anticipate the type of objects it needs to create.
    - C) When you need to create complex objects with multiple parts.
    - D) When you want to create multiple instances of a class.
    - **Answer: B**

39. **Which of the following is NOT a benefit of the Factory Method pattern?**
    - A) It promotes loose coupling.
    - B) It encapsulates object creation.
    - C) It restricts object creation to one instance.
    - D) It allows subclasses to decide which objects to create.
    - **Answer: C**

40. **Which pattern allows adding new product types without modifying existing code?**
    - A) Singleton
    - B) Factory Method
    - C) Abstract Factory
    - D) Builder
    - **Answer: B**

Abstract Factory Advanced

41. **In the Abstract Factory pattern, the `ConcreteFactory` class is responsible for:**
    - A) Creating families of related products.
    - B) Cloning objects.
    - C) Creating a single object.
    - D) Implementing a global access point.

- **Answer: A**

42. **Which of the following is an advantage of the Abstract Factory pattern?**
    - A) It simplifies the process of object creation.
    - B) It enforces the creation of a single instance.
    - C) It isolates clients from concrete classes.
    - D) It simplifies the process of cloning objects.
    - **Answer: C**

43. **What is the relationship between Abstract Factory and Factory Method patterns?**
    - A) Abstract Factory uses Factory Methods to create families of products.
    - B) Factory Method uses Abstract Factory to create a single product.
    - C) They are unrelated patterns.
    - D) They both enforce singleton-like behavior.
    - **Answer: A**

44. **Which of the following scenarios is best suited for the Abstract Factory pattern?**
    - A) Creating an instance of a single class.
    - B) Creating multiple related objects that must work together.
    - C) Cloning an object with multiple configurations.
    - D) Managing a global state.
    - **Answer: B**

45. **Which component is NOT part of the Abstract Factory pattern?**
    - A) Abstract Factory
    - B) Concrete Factory
    - C) Product
    - D) Singleton
    - **Answer: D**

Builder Advanced

46. **The Builder pattern is useful when:**
    - A) An object needs to be created in one step.
    - B) An object needs to be created with a varying set of configurations.
    - C) You need to restrict the object creation to one instance.
    - D) You need to clone an existing object.
    - **Answer: B**

47. **In the Builder pattern, which class is responsible for managing the construction process?**
    - A) Builder
    - B) ConcreteBuilder
    - C) Director
    - D) Product
    - **Answer: C**

48. **What is a common use case for the Builder pattern?**
    - A) When creating simple objects.
    - B) When creating complex objects with many optional parts.
    - C) When cloning objects.
    - D) When creating global objects.
    - **Answer: B**

49. **Which of the following is NOT an advantage of the Builder pattern?**
    - A) It provides better control over the construction process.
    - B) It allows the creation of immutable objects.
    - C) It simplifies the creation of complex objects.
    - D) It enforces single instance creation.
    - **Answer: D**

50. **The `Product` class in the Builder pattern is:**
    - A) The object being built.

- B) The object that manages the construction process.
- C) The interface that defines the building steps.
- D) The object that decides how the product is created.
- **Answer: A**

Prototype Advanced

51. **The Prototype pattern is particularly useful when:**
- A) Creating a single instance of a class.
- B) Creating new objects by copying existing ones.
- C) Creating families of related objects.
- D) Managing complex construction processes.
- **Answer: B**

52. **Which method in C# is commonly associated with the Prototype pattern?**
- A) `FactoryMethod`
- B) `Build`
- C) `Clone`
- D) `GetInstance`
- **Answer: C**

53. **A shallow copy created by `MemberwiseClone` in C# does NOT:**
- A) Create a new instance of the object.
- B) Copy reference types as new instances.
- C) Copy value types.
- D) Create a new memory address for the cloned object.
- **Answer: B**

54. **What is the primary difference between a shallow copy and a deep copy in the Prototype pattern?**
- A) A shallow copy clones all the fields and objects, while a deep copy does not.

- B) A deep copy clones all fields and objects, while a shallow copy only copies the top-level object.
- C) A deep copy does not clone reference types.
- D) A shallow copy does not clone value types.
- **Answer: B**

55. **Which of the following is a disadvantage of the Prototype pattern?**
- A) It is difficult to implement in object-oriented languages.
- B) It can be less efficient if deep copies are needed frequently.
- C) It enforces the use of private constructors.
- D) It requires global access points for object creation.
- **Answer: B**

Mixed Advanced

56. **Which of the following design patterns would you use to create an object that should only have one instance?**
- A) Singleton
- B) Factory Method
- C) Prototype
- D) Abstract Factory
- **Answer: A**

57. **Which pattern would you choose to clone an object rather than create a new one?**
- A) Singleton
- B) Factory Method
- C) Prototype
- D) Builder
- **Answer: C**

58. **When dealing with complex object construction where different configurations are required, which pattern is most appropriate?**
- A) Singleton

- B) Factory Method
- C) Abstract Factory
- D) Builder
- **Answer: D**

59. **Which design pattern helps to create families of related objects without specifying their concrete classes?**
- A) Singleton
- B) Abstract Factory
- C) Prototype
- D) Builder
- **Answer: B**

60. **If a client needs to be isolated from the concrete classes that create objects, which pattern should be used?**
- A) Singleton
- B) Factory Method
- C) Abstract Factory
- D) Prototype
- **Answer: C**

Singleton Pattern Specific

61. **Which of the following statements about the Singleton pattern is incorrect?**
- A) Singleton can be lazy-loaded.
- B) Singleton cannot be used in a multithreaded environment.
- C) Singleton provides a global access point to an instance.
- D) Singleton can be combined with the Factory Method pattern.
- **Answer: B**

62. **What ensures that only one instance of a Singleton class is created in a multithreaded environment?**

- A) The `static` keyword.
- B) A `lock` statement.
- C) The `readonly` keyword.
- D) Making the constructor `private`.
- **Answer: B**

63. **Which of the following techniques is used to make a Singleton pattern thread-safe without using `lock`?**

- A) Double-checked locking.
- B) Eager initialization.
- C) Lazy initialization.
- D) Using a `volatile` keyword.
- **Answer: A**

64. **How do you prevent instantiation of a Singleton class using reflection?**

- A) Throw an exception in the constructor if an instance already exists.
- B) Use `sealed` keyword.
- C) Use the `lock` keyword.
- D) Use the `private` keyword in the constructor.
- **Answer: A**

65. **Which is NOT a typical use case for the Singleton pattern?**

- A) Logging service.
- B) Configuration manager.
- C) Database connection pool.
- D) Factory for creating objects.
- **Answer: D**

Factory Method Pattern Specific

66. **In the Factory Method pattern, how does the `Creator` class handle the object creation process?**

- A) It delegates object creation to a subclass.
- B) It creates the object itself.
- C) It uses reflection to instantiate the object.
- D) It only allows one instance to be created.
- **Answer: A**

67. **Which of the following is a characteristic of the Factory Method pattern?**

- A) It returns an instance of a single class.
- B) It provides a way to delegate the creation of objects to subclasses.
- C) It ensures only one instance of an object is created.
- D) It clones an existing object.
- **Answer: B**

68. **In the Factory Method pattern, the `Product` interface is responsible for:**

- A) Defining the operations that all concrete products must implement.
- B) Creating the product itself.
- C) Managing the construction process.
- D) Ensuring that only one product is created.
- **Answer: A**

69. **Which of the following is NOT a benefit of using the Factory Method pattern?**

- A) Code is more maintainable and extensible.
- B) Object creation is centralized.
- C) Tight coupling between Creator and Product classes.
- D) New product types can be added without modifying existing code.
- **Answer: C**

70. **What is a common use case for the Factory Method pattern?**

- A) Managing a global state.

- B) Creating a single instance of a class.
- C) Creating a complex object with multiple parts.
- D) Deferring the instantiation to subclasses.
- **Answer: D**

Abstract Factory Pattern Specific

71. **Which of the following is an example of when to use the Abstract Factory pattern?**
- A) When you want to create a single object.
- B) When you want to create objects without specifying their concrete types.
- C) When you want to ensure that only one instance of an object is created.
- D) When you want to clone an existing object.
- **Answer: B**

72. **The Abstract Factory pattern is designed to create:**
- A) Single objects.
- B) Families of related objects.
- C) Clones of existing objects.
- D) Immutable objects.
- **Answer: B**

73. **Which of the following classes typically provides the interface for creating families of related objects in the Abstract Factory pattern?**
- A) Concrete Factory
- B) Product
- C) Abstract Factory
- D) Prototype
- **Answer: C**

74. **What does the `ConcreteFactory` class do in the Abstract Factory pattern?**
- A) It implements the interface for creating products.

- B) It defines the product interfaces.
- C) It ensures that only one product is created.
- D) It clones an existing object.
- **Answer: A**

75. **Which of the following is NOT typically a product created by an Abstract Factory?**
- A) A graphical user interface widget.
- B) A database connection.
- C) A single instance of a logger.
- D) A set of related objects like buttons and text boxes.
- **Answer: C**

Builder Pattern Specific

76. **Which of the following best describes the role of the `Director` in the Builder pattern?**
- A) The Director defines the construction process for complex objects.
- B) The Director constructs the final object directly.
- C) The Director provides the global access point to the object.
- D) The Director clones an existing object.
- **Answer: A**

77. **Which of the following is NOT an advantage of using the Builder pattern?**
- A) Allows different representations of the same construction process.
- B) Constructs objects in a step-by-step manner.
- C) Ensures a single instance of an object.
- D) Provides better control over the construction process.
- **Answer: C**

78. **Which part of the Builder pattern actually assembles the parts to construct the final object?**
- A) Director
- B) Builder

- C) ConcreteBuilder
- D) Product
- **Answer: C**

79. **Which scenario is best suited for using the Builder pattern?**
    - A) When constructing simple objects with a single method.
    - B) When constructing complex objects with many configurations.
    - C) When cloning an existing object.
    - D) When managing a global state.
    - **Answer: B**

80. **Which of the following is true about the `Product` in the Builder pattern?**
    - A) It is the complex object that is being constructed.
    - B) It is responsible for managing the construction process.
    - C) It defines the steps required to construct the object.
    - D) It enforces that the object is only created once.
    - **Answer: A**

Prototype Pattern Specific

81. **Which method is essential for implementing the Prototype pattern in C#?**
    - A) `Clone`
    - B) `FactoryMethod`
    - C) `Build`
    - D) `GetInstance`
    - **Answer: A**

82. **Which type of copy is created by the `MemberwiseClone` method in C#?**
    - A) Deep copy
    - B) Shallow copy
    - C) Immutable copy

- D) Singleton copy

- **Answer: B**

83. **What is the key difference between shallow copy and deep copy in the context of the Prototype pattern?**

- A) Shallow copy clones only primitive types, deep copy clones reference types as well.

- B) Shallow copy clones the entire object graph, deep copy does not.

- C) Deep copy clones the entire object graph, shallow copy does not.

- D) Shallow copy clones reference types as new objects, deep copy does not.

- **Answer: C**

84. **Which of the following is NOT an advantage of using the Prototype pattern?**

- A) It reduces the need for creating instances from scratch.

- B) It allows for quick creation of new objects based on existing ones.

- C) It simplifies the object creation process in multithreaded environments.

- D) It can reduce the performance cost associated with object creation.

- **Answer: C**

85. **In which scenario would you prefer the Prototype pattern over the Factory Method pattern?**

- A) When object creation is simple and direct.

- B) When creating objects with complex configurations.

- C) When you need to create multiple objects that share the same configuration.

- D) When object creation needs to be deferred to subclasses.

- **Answer: C**

Mixed Advanced

86. **Which pattern would you use if you need to ensure that only one instance of a configuration class is created throughout the application?**

- A) Singleton

- B) Factory Method

- C) Prototype

- D) Builder
- **Answer: A**

87. **Which pattern would you choose to create a complex object with different configurations depending on the client's needs?**
- A) Singleton
- B) Builder
- C) Abstract Factory
- D) Prototype
- **Answer: B**

88. **Which pattern would you use to create families of related objects without specifying their concrete classes?**
- A) Singleton
- B) Factory Method
- C) Abstract Factory
- D) Prototype
- **Answer: C**

89. **Which pattern allows for the creation of new objects by copying an existing object, rather than creating a new instance?**
- A) Singleton
- B) Factory Method
- C) Abstract Factory
- D) Prototype
- **Answer: D**

90. **In a system where different types of objects need to be created dynamically based on the input, which pattern is most suitable?**
- A) Singleton
- B) Factory Method
- C) Abstract Factory

- D) Builder

- **Answer: B**

Singleton Pattern Expert Level

91. **Which of the following implementations of Singleton in C# is thread-safe by default without using `lock`?**

- A) Lazy initialization

- B) Eager initialization

- C) Double-checked locking

- D) `Lazy<T>` type

- **Answer: D**

92. **How can a Singleton instance be garbage collected in C#?**

- A) By ensuring no references to it exist.

- B) By using the `lock` keyword.

- C) By overriding the `Finalize` method.

- D) Singleton instances cannot be garbage collected.

- **Answer: A**

93. **What is the primary benefit of using the `Lazy<T>` type for Singleton implementation in C#?**

- A) It ensures thread-safety without using locks.

- B) It makes the Singleton eagerly initialized.

- C) It allows multiple instances.

- D) It simplifies object cloning.

- **Answer: A**

94. **Which of the following Singleton implementations is least efficient in a high-concurrency scenario?**

- A) `Lazy<T>` type

- B) Double-checked locking

- C) Eager initialization

- D) Simple lock-based initialization

- **Answer: D**

95. **In a multithreaded environment, which Singleton implementation would you choose for simplicity and thread-safety?**

- A) Simple lock-based initialization

- B) Double-checked locking

- C) `Lazy<T>` type

- D) Eager initialization

- **Answer: C**

Factory Method Pattern Expert Level

96. **Which pattern is the Factory Method pattern often confused with due to its similarity in purpose?**

- A) Abstract Factory

- B) Singleton

- C) Prototype

- D) Builder

- **Answer: A**

97. **In a Factory Method pattern, what is a disadvantage of creating concrete product classes directly in the Creator class?**

- A) It reduces the flexibility of adding new product types.

- B) It increases the complexity of the code.

- C) It enforces tight coupling between classes.

- D) It reduces code readability.

- **Answer: A**

98. **Which of the following is NOT typically achieved by the Factory Method pattern?**

- A) Decoupling of object creation from the client.

- B) Centralized object creation.

- C) Single responsibility principle.
- D) Creation of complex objects.
- **Answer: D**

99. **How does the Factory Method pattern contribute to the open/closed principle?**
- A) By allowing new products to be added without changing existing code.
- B) By ensuring that only one instance of an object is created.
- C) By providing a global access point to objects.
- D) By enforcing a specific construction sequence.
- **Answer: A**

100. **In the Factory Method pattern, what role does the Creator class play?**
- A) It delegates the object creation to concrete classes.
- B) It defines the interface for creating an object.
- C) It implements the creation logic for a specific object.
- D) It ensures thread safety during object creation.
- **Answer: B**