**Definition, place of design pattern in a world of other patterns**

Test:

1. What is the best definition of pattern?

A. Pattern - is the set of instructions that need to be done in order to solve a problem.

B. Pattern - is a universal solution that can solve a vast majority of problems.

C. Pattern - is a solution to a specific type of problems that occur over and over again within a given context.  
  
**Answer: C**

2. What is the correct description of software design patterns?

A. Software design pattern - is set of patterns which solve problems related to integration of new and existing software in a business environment.

B. Software design pattern - is set of patterns which solve commonly occurring problem within a given context in software design, which typically show relationships and interactions between classes or objects.

C. Software design pattern - is set of patterns which solve commonly occurring problem in software architecture within a given context.

**Answer: C**

3. Which of the following is true about design patterns?

A. Design patterns represent the best practices used by experienced object-oriented software developers.

B. Design patterns are solutions to general problems that software developers faced during software development.

C. Design patterns are obtained by trial and error by numerous software developers over quite a substantial period of time.

D. All of the above.

**Answer: D**

**Creational patterns: definitions, problem and solution statements**

Test:

1. What is the best definition of Creational patterns?

A. This type of patterns provides a way to create interfaces and define ways to compose objects to obtain new functionalities.

B. This type of patterns provides an ability to hide creation logic instead of using new operator.

C. This type of patterns is specifically concerned with communication between objects.

D. This type of patterns is specifically concerned with the presentation tier.

**Answer: B**

2. What is related to Creational patterns? Multiple options available.

A. Creational patterns are concerned with how objects and classes communicate with each other.

B. Creational patterns hide how instances of classes are created and put together.

C. Creational patterns describe ways to compose objects to realize new functionality.

D. Creational patterns show ways how classes and objects are composed to form larger structures.

E. Creational patterns encapsulate knowledge about which concrete classes the system uses.

**Answer: B, E**

3. What from the following code issues can be solved with Creational Patterns? Multiple answers possible.

A. Algorithm implementation contains too many special case logic and conditional statements.

B. Different methods in subclass do semantically similar steps except for creating objects.

C. Creation code is duplicated in different methods.

D. Class has new responsibility which is additional to its base responsibility which makes the class very big.

E. Creation logic is sprawled among many classes.

F. Classes implement the same of similar steps and have different interface which make client code complicated, since it has to work with both interfaces.

**Answer: B, C**

4. Which of the following pattern creates object without exposing the creation logic to the client and refer to newly created object using a common interface?

A. Factory Pattern.

B. Abstract Factory Pattern.

C. Singleton Pattern.

D. Prototype Pattern.

**Answer: A**

5. Which of the following describes the Abstract Factory pattern correctly?

A. This pattern creates object without exposing the creation logic to the client and refer to newly created object using a common interface.

B. In this pattern an interface is responsible for creating a factory of related objects without explicitly specifying their classes.

C. This pattern involves a single class which is responsible to create an object while making sure that only single object gets created.

**Answer: B**

6. Which Design Pattern should you use when a class wants its subclasses to specify the objects it creates?

A. Abstract Factory.

B. Builder.

C. Factory Method.

D. Singleton.

**Answer: C**

7. Which Design Pattern should you use when there must be exactly one instance of a class, and it must be accessible to clients from a well-known access point.

A. Abstract Factory.

B. Builder.

C. Factory Method.

D. Singleton.

**Answer: D**

8. Which Design Pattern should you use when a system should be configured with one of multiple families of products.

A. Builder.

B. Abstract Factory.

C. Factory Method.

D. Singleton.

**Answer: B**

**Structural patterns: definitions, problem and solution statements**

Test:

1. What is the best definition of Structural patterns?

A. This type of patterns provides a way to create interfaces and define ways to compose objects to obtain new functionalities.

B. This type of patterns provides an ability to hide creation logic instead of using new operator.

C. This type of patterns is specifically concerned with communication between objects.

D. This type of patterns is specifically concerned with the presentation tier.

**Answer: A**

2. What is related to Structural patterns? Multiple options available.

A. Structural patterns are concerned with how objects and classes communicate with each other.

B. Structural patterns hide how instances of classes are created and put together.

C. Structural patterns describe ways to compose objects to realize new functionality.

D. Structural patterns show ways how classes and objects are composed to form larger structures.

E. Structural patterns encapsulate knowledge about which concrete classes the system uses.

F. Structural patterns describe how different objects work together to accomplish a task.

**Answer: D, C, E**

3. What from the following code issues can be solved with Structural Patterns? Multiple answers possible.

A. Algorithm implementation contains too many special case logic and conditional statements.

B. Different methods in subclass do semantically similar steps except for creating objects.

C. Creation code is duplicated in different methods.

D. Class contains hard-coded logic to notify other classes.

E. Creation logic is sprawled among many classes.

F. Classes implement the same of similar steps and have different interface which make client code complicated, since it has to work with both interfaces.

G. Class has new responsibility which is additional to its base responsibility which makes the class very big.

**Answer: E, F, G**

4. Which Design Pattern should you use when you want to avoid a permanent binding between an abstraction and its implementation. This might be the case, for example, when the implementation must be selected or switched at run-time.

A. Adapter.

B. Composite.

C. Bridge.

D. Decorator.

**Answer: C**

5. Which Design Pattern should you use when you want to represent part-whole hierarchies of objects.

A. Adapter.

B. Composite.

C. Bridge.

D. Decorator.

**Answer: C**

6. Which Design Pattern should you use when you want to provide a simple interface to a complex subsystem.

A. Adapter.

B. Façade.

C. Decorator.

D. Composite.

**Answer: B**

7. Which Design Pattern should you use when an application uses a large number of objects and the storage costs are high because of the sheer quantity of objects.

A. Adapter.

B. Façade.

C. Flyweight.

D. Decorator.

**Answer: C**

8. Which Design Pattern should you use when you want to add responsibilities to individual objects dynamically and transparently, that is, without affecting other objects.

A. Adapter.

B. Façade.

C. Decorator.

D. Composite.

**Answer: C**

**Behavioral patterns: definitions, problem and solution statements**

Test:

1. What is the best definition of Behavioral patterns?

A. This type of patterns provides a way to create interfaces and define ways to compose objects to obtain new functionalities.

B. This type of patterns provides an ability to hide creation logic instead of using new operator.

C. This type of patterns is specifically concerned with communication between objects.

D. This type of patterns is specifically concerned with the presentation tier.

**Answer: C**

2. What is related to Behavioral patterns? Multiple options available.

A. Behavioral patterns are concerned with how objects and classes communicate with each other.

B. Behavioral patterns hide how instances of classes are created and put together.

C. Behavioral patterns describe ways to compose objects to realize new functionality.

D. Behavioral patterns show ways how classes and objects are composed to form larger structures.

E. Behavioral patterns encapsulate knowledge about which concrete classes the system uses.

F. Behavioral patterns describe how different objects work together to accomplish a task.

**Answer: A, F**

3. What from the following code issues can be solved with Behavioral Patterns? Multiple answers possible.

A. Algorithm implementation contains too many special case logic and conditional statements.

B. Different methods in subclass do semantically similar steps except for creating objects.

C. Creation code is duplicated in different methods.

D. Class contains hard-coded logic to notify other classes.

E. Creation logic is sprawled among many classes.

F. Classes implement the same of similar steps and have different interface which make client code complicated, since it has to work with both interfaces.

G. Class has new responsibility which is additional to its base responsibility which makes the class very big.

**Answer: A, D**

4. Which of the following describes the Command pattern correctly?

A. This pattern is used to get a way to access the elements of a collection object in sequential manner without any need to know its underlying representation.

B. This pattern creates a chain of receiver objects for a request.

C. This pattern provides a way to evaluate language grammar or expression.

D. In this pattern a request is wrapped under an object as command and passed to invoker object.

**Answer: D**

5. Which Design Pattern should you use when many related classes differ only in their behavior or you need different variants of an algorithm?

A. Command.

B. Strategy.

C. Iterator.

D. Observer.

**Answer: B**

6. Which Design Pattern should you use when you want to access an aggregate object's contents without exposing its internal representation? Aggregate object means an object, which contains a collection of other objects.

A. Template Method

B. Iterator.

C. Strategy.

D. Mediator.

**Answer: B**

7. Which Design Pattern should you use when there is a language to interpret, and you can represent statements in the language as abstract syntax trees.

A. Mediator.

B. Strategy.

C. Interpreter.

D. Chain of responsibility.

**Answer: C**

8. Which Design Pattern should you use when a set of objects communicate in well-defined but complex ways. The resulting interdependencies are unstructured and difficult to understand.

A. Adapter.

B. Mediator.

C. Template Method.

D. Iterator.

**Answer: B**

9. Which Design Pattern should you use when an object should be able to notify other objects without making assumptions about who these objects are. In other words, you don't want these objects tightly coupled.

A. Command.

B. Adapter.

C. Observer.

D. Chain of Responsibility.

**Answer: C**

10. Which Design Pattern should you use when more than one object may handle a request, and the handler is not known a priori. The handler should be ascertained automatically.

A. Observer.

B. Chain of Responsibility.

C. Adapter.

D. Mediator.

**Answer: B**

11. Which Design Pattern should you use when an object's behavior depends on its state, and it must change its behavior at run-time depending on that state.

A. Mediator.

B. Adapter.

C. State.

D. Observer.

**Answer: C**