**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.

**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.

**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.

**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.

**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.

**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.

**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.

**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.

**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.

**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.

**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.