1. **What is the best definition of a pattern?**   
   A. Pattern - is the set of instructions that need to be followed 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 occurs repeatedly within a given context.
2. **What is the correct description of software design patterns?**   
   A.Software design patterns - are solutions which solve problems related to the integration of new and existing software in a business environment.  
   B. Software design patterns - are solutions which solve commonly occurring problems within a given context in software design, which typically show relationships and interactions between classes or objects.  
   C. Software design patterns - are solutions which solve commonly occurring problems in software architecture within a given context.
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 through trial and error by numerous software developers over quite a substantial period of time.  
   D. All the above.
4. **What is the best definition of Behavioral patterns?**   
   A. These types of patterns provide a way to create interfaces and define ways to compose objects to obtain new functionalities.  
   B. These types of patterns provide an ability to hide creation logic instead of using new operator.  
   C. These types of patterns are specifically concerned with communication between objects.  
   D. These types of patterns are specifically concerned with the presentation tier.
5. **Which of the following options are related to Behavioral patterns? Multiple choices 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 the ways to compose objects to realize a new functionality.  
   D. Behavioral patterns show how classes and objects are composed to form larger structures.  
   E. Behavioral patterns encapsulate knowledge of the exact classes the system uses.  
   F. Behavioral patterns describe how different objects work together to accomplish a task.
6. **Which issues from the following code can be solved with Behavioral Patterns? Multiple answers are possible.**   
   A. Algorithm implementation contains too many  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 steps and have different interfaces -  this makes client code complicated, because it must work with both interfaces.  
   G. Class has a new responsibility which is additional to its base responsibility which makes the class very big.
7. **Which of the following describe the Command pattern correctly?**   
   A. This pattern is used to access the elements of a collection object in a 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 around an object as a command and passed to invoker object.
8. **Which Design Pattern should you use when many related classes differ only in their behavior, or do you need different variants of an algorithm?**   
   A. Command.  
   B. Strategy.  
   C. Iterator.  
   D. Observer.
9. **Which Design Pattern should you use when you want to access aggregate object's contents without exposing its internal representation? An aggregate object is an object, which contains a collection of other objects.**   
   A. Template Method  
   B. Iterator.  
   C. Strategy.  
   D. Mediator.
10. **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.
11. **Which Design Pattern should you use when 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.
12. **Which Design Pattern should you use when an object can notify other objects without making assumptions about who these objects are? In other words, you do not want these objects to be tightly coupled.**   
    A. Command.  
    B. Adapter.  
    C. Observer.  
    D. Chain of Responsibility.
13. **Which Design Pattern should you use when there is more than one object that can 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.
14. **Which Design Pattern should you use when 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.
15. **What is the best definition of Creational patterns?**   
    A. These types of patterns provide a way to create interfaces and define ways to compose objects to obtain new functionalities.  
    B. These types of patterns provide an ability to hide creation logic instead of using a new operator.  
    C. These types of patterns are specifically concerned with communication between objects.  
    D. These types of patterns are specifically concerned with the presentation tier.
16. **Which of the following options are related to Creational patterns? Multiple choices 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 the ways to compose objects to realize a new functionality.  
    D. Creational patterns show how classes and objects are composed to form larger structures.  
    E. Creational patterns encapsulate knowledge of the exact classes the system uses.
17. **Which issues from the following code can be solved with Creational Patterns? Multiple answers are possible.**   
    A. Algorithm implementation contains too many 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 a 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 steps and have different interfaces   - this makes client code complicated, because it must work with both interfaces.
18. **Which pattern creates an object without exposing the creation logic to the client and refers to the newly created object using a common interface?**   
    A. Factory Pattern.  
    B. Abstract Factory Pattern.  
    C. Singleton Pattern.  
    D. Prototype Pattern.
19. **Which statement describe the Abstract Factory pattern correctly?**   
    A. This pattern creates an object without exposing the creation logic to the client and refers to the 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 for creating an object while making sure that only a single object gets created.
20. **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.
21. **When there must be exactly one instance of a class, and it must be accessible to clients from a well-known access point which Design Pattern should you use?**     
    A. Abstract Factory.  
    B. Builder.  
    C. Factory Method.  
    D. Singleton.
22. **Which Design Pattern should you use when a system should be configured with one of the multiple families of products?**   
    A. Builder.  
    B. Abstract Factory.  
    C. Factory Method.  
    D. Singleton.
23. **What is the best definition of Structural patterns?**   
    A. These types of patterns provide a way to create interfaces and define ways to compose objects to obtain new functionalities.  
    B. These types of patterns provide an ability to hide creation logic instead of using new operator.  
    C. These type of patterns are specifically concerned with communication between objects.  
    D. These type of patterns are specifically concerned with the presentation tier.
24. **Which of the following options are related to Structural patterns? Multiple choices 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 the ways to compose objects to realize a new functionality.  
    D. Structural patterns show how classes and objects are composed to form larger structures.  
    E. Structural patterns encapsulate knowledge of the exact classes the system uses.  
    F. Structural patterns describe how different objects work together to accomplish a task.
25. **Which of the issues from the following code can be solved with Structural Patterns? Multiple answers are possible.**   
    A. Algorithm implementation contains too many 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 steps and have different interfaces - this makes client code complicated, because it must work with both interfaces.  
    G. Class has a new responsibility which is additional to its base responsibility which makes the class very big.
26. **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.
27. **Which Design Pattern should you use when you want to represent part-whole hierarchies of objects?**   
    A. Adapter.  
    B. Composite.  
    C. Bridge.  
    D. Decorator.
28. **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.
29. **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.
30. **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.