

Test 2

1. Given variables I, J, and K at the end of the given code segment, which of the following assertions are true? (select all correct answers). $J = I$; **if** ($K \geq J$) $J = K$;
 - a. $J \leq I$ and $J \leq K$
 - b. I is unchanged
 - c. $J \geq I$ and $J \geq K$
 - d. J is unchanged
 - e. K is unchanged
2. A(n) _____ is a general reusable solution to a commonly occurring problem within a given context in software design. We may use several of them in one system
 - a. Framework
 - b. Library
 - c. Design pattern
 - d. Architectural pattern
3. Which of the following is true about hook methods in the template method pattern
 - a. Can be implemented in a java interface
 - b. Can get access to private data through other methods
 - c. Needs access to the private data of a class
 - d. Declared as abstract
 - e. Not abstract
4. Which of the following are advantages of using Blackbox testing
 - a. You do not need to rewrite test cases after the code has changed as long as the interface and contracts have not changed
 - b. You can guarantee 100% code coverage
 - c. Test cases can be reused for any implementation of the same interface
 - d. You can write test cases before the code is finished
5. A(n) _____ is a solution for the entire structure of a program. We would only use one in our system
 - a. Design pattern

- b. Library
 - c. Architectural pattern
 - d. Framework
6. A(n) _____ is a set of specific test cases we create to test out component
- a. JUnit framework
 - b. Test stub
 - c. Test driver
 - d. Test plan
7. Which of the following are an example of a Design Pattern?
- a. Polling
 - b. Model view controller
 - c. Factory method
 - d. Observer
 - e. Template hook
 - f. Decorator
8. What is needed to have a complete test case
- a. Test stub
 - b. Input
 - c. The JUnit test framework
 - d. Fault
 - e. Failure
 - f. Expected output
9. Which of the following is true about template methods in the template method pattern
- a. Declared as abstract
 - b. Not abstract
 - c. Needs access to the private data of a class
 - d. Can be implemented in a java interface
 - e. Can get access to private data through other methods
10. We use the _____ tag in JUnit to mark a function as a test case that should be run

- a. @TestCase
 - b. @Unit
 - c. @Test
 - d. @JUnit
 - e. @TestPlan
11. We can use the _____ tag to force our JUnit tests to stop and be marked as a failure if the test case takes too long
- a. @timeout
 - b. @timed
 - c. @limit
 - d. @loopcheck
12. If we are trying to write a proof to verify that our public method does not contain faults, what will we try to prove to be true at the end of our proof?
- a. The preconditions of our method
 - b. The invariants of the class that contains our method
 - c. The postconditions of our method
 - d. We cannot assume anything to be true
13. Which of the following are advantages of using Whitebox testing
- a. You can write test cases before the code is finished
 - b. You can guarantee 100% code coverage
 - c. Test cases can be reused for any implementation of the interface
 - d. You do not need to rewrite test cases after the code has changed
14. Our program is in a(n) _____ if any further processing will result in a failure.
- a. Erroneous state
15. A(n) _____ is the mechanical or algorithmic cause of the failure.
- a. fault
16. A(n) _____ is any deviation of the observed output from the expected or specified output.
- a. failure
17. Consider the following strings:

s = <9, 3, 2, 3>

t = <41, 8, 4, 4, 2, 4, 1, 8>

u = <6, 3, 8, 4, 6>

What is the Occurs_Ct(2, s)?

1

18. Consider the following strings:

s = <19, 4, 8, 1, 4>

t = <7, 9, 4, 4>

u = <5, 4, 8, 7, 5, 4>

What is Prt_Btwn(3, 6, u)?

Note: Prt_Btwn(n, m, string)

n is inclusive, m is exclusive, positioning starts at 0.

<7, 5, 4>

19. Assume s is a Stack of integers with maximum depth 3. The abstraction of stack is a string of integers and the specification of pop operation is given below.

```
/**  
 * @pre length of self, |self| > 0  
 * @post self = pop() removes and returns the front element from #self  
 */  
public T pop( );
```

Which of the following are valid test inputs? Check all correct ones.

a. S = <1, 2>

b. S = <>

c. S = <1, 2, 3>

d. `S = empty_string`

20. Consider a system for a pizzeria that would allow people to order pizzas online. This system follows the Model View Controller architectural pattern. For each provided class, read its description and indicate what layer of the architecture that class would belong to.

A class called `GenerateReport` that contains the code to handle the process of generating a daily end-of-day report. The class does not contain the actual information of the report, just the process.

a. Controller

b. Model

c. View

21. Consider a system for a pizzeria that would allow people to order pizzas online. This system follows the Model View Controller architectural pattern. For each provided class, read its description and indicate what layer of the architecture that class would belong to.

A class called `Order` that would hold the information regarding an order (pizzas on the order, sides, etc.)

a. Model

b. View

c. Controller

22. Consider a system for a pizzeria that would allow people to order pizzas online. This system follows the Model View Controller architectural pattern. For each provided class, read its description and indicate what layer of the architecture that class would belong to.

A class called `ViewInventory` that displays the current items in our inventory to the user and allows them to select inventory items to get more information

a. Model

b. View

c. Controller

23. You will have a data structure that is used frequently in your system (such as a `List`) and you would like to be able to easily change the implementation of that data structure in

your system without having to replace every constructor call. Which design pattern would you use?

- a. Decorator
- b. Factory Method**
- c. Model View Controller
- d. Template Hook
- e. Polling
- f. Observer

24. You have an interface, such as a List, that could have many possible implementations.

Some methods need direct access to private data of the implementation to work, but not all of them. Which design pattern would you use to save some time and effort?

- a. Decorator
- b. Observer
- c. Polling
- d. Model View Controller
- e. Factory Method
- f. Template Hook**

25. You have an existing interface and implementation. You want to be able to add some new functionality to the interface but you are not allowed to edit the existing files. You create an extended interface to add in the new functionality, but now you need a new implementation as well. You do not want to have to re-code all the methods from the original interface and implementation.

- a. Observer
- b. Template hook
- c. Factory method
- d. Decorator**
- e. Polling
- f. Model view controller

26. You are creating a JUnit test suite for your interface. Each test case creates a new object that implements that interface. You would like to be able to change which

implementation of the interface is being used without having to change every JUnit test case. Which design pattern would you use?

- a. Observer
- b. Template hook
- c. Factory method
- d. Decorator
- e. Polling
- f. Model view controller

27. Consider a class that implements a user interface called UserScreen and a class called Widget that represents objects on the screen the user can interact with. UserScreen does not know which Widgets the user will interact with when they will interact with them, or in what order. UserScreen still needs to know when a user interacts with a Widget so it can react appropriately to it. You know you will need to use the **Observer Pattern** to make this work. Answer the following questions about how you can make your current classes of UserScreen and Widget follow the **Observer Pattern**.

What class or classes will play the role of the OBSERVER?

UserScreen

28. Consider a class that implements a user interface called UserScreen and a class called Widget that represents objects on the screen the user can interact with. UserScreen does not know which Widgets the user will interact with when they will interact with them, or in what order. UserScreen still needs to know when a user interacts with a Widget so it can react appropriately to it. You know you will need to use the **Observer Pattern** to make this work. Answer the following questions about how you can make your current classes of UserScreen and Widget follow the **Observer Pattern**.

What additional method(s) and/or private variables will UserScreen need to provide to play its role in the Observer Pattern?

UserScreen will need to provide a callback method in order to ask for updates on Widgets

29. Consider a class that implements a user interface called `UserScreen` and a class called `Widget` that represents objects on the screen the user can interact with. `UserScreen` does not know which `Widgets` the user will interact with when they will interact with them, or in what order. `UserScreen` still needs to know when a user interacts with a `Widget` so it can react appropriately to it. You know you will need to use the **Observer Pattern** to make this work. Answer the following questions about how you can make your current classes of `UserScreen` and `Widget` follow the **Observer Pattern**.

How will a `Widget` object respond to a user event, such as the user clicking on that `Widget`?

It will call the callback method with an update and send it back to the observer

30. Generic data types require that the data type passed in as the parameterized data type is a reference data type. Why is this required for generics to work and what can we expect about these types in Java?
- Generic types just manipulate pointer, that's why they need to be a reference type and we expect that these types will extend the `Object` class and be able to use `toString()` and `equal()` methods and have casting abilities.