## Help & Tutorials



## Online Training by STH

# White Box Testing: A Complete Guide with Techniques, Examples, & Tools

Posted In | Testing Concepts, Testing Methodologies, Types of Testing | Last Updated: "June 11, 2018"

**What is White Box Testing?**



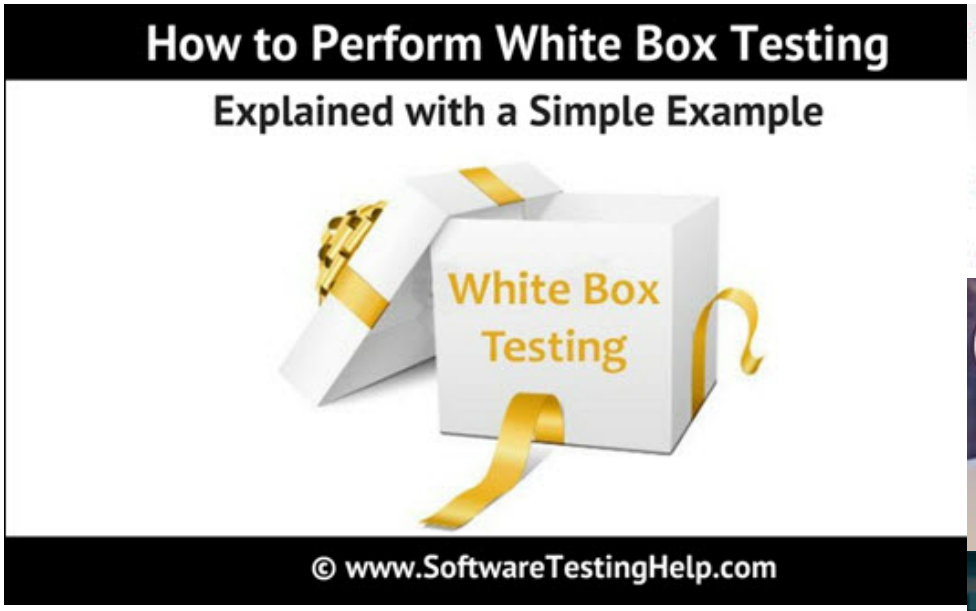
If we go by the definition, "White box testing" (also known as clear, glass box or structural testing) is a testing technique which evaluates the code and the internal structure of a program.

White box testing involves looking at the structure of the code. When you know the internal structure of a product, tests can be conducted to ensure that the internal operations performed according to the specification. And all internal components have been adequately exercised.

## People Also Read..



## Adv










**What You Will Learn:** [show]

## My Experience

It's almost a decade now since I'm into Software testing field and so far noticed that the testers are the most enthusiastic in the whole software industry.

The prime reason behind this is – tester always has something in their scope to learn. Be it a domain, process or a technology, a tester can have a complete development if they wish to.

But as they say *"There is always a darker side"*.

Testers also indeed avoid a type of testing which they feel to be very complicated and the developer's piece of cake. Yes, the "White Box

Testing”.



## Coverage

**White Box Testing is coverage of the specification in the code:**

**1. Code coverage**

**2. Segment coverage:** Ensure that each code statement is executed once.

**3. Branch Coverage or Node Testing:** Coverage of each code branch in from all possible was.

**4. Compound Condition Coverage:** For multiple conditions test each condition with multiple paths and combination of the different path to reach that condition.

**5. Basis Path Testing:** Each independent path in the code is taken for testing.

**6. Data Flow Testing (DFT):** In this approach you track the specific variables through each possible calculation, thus defining the set of intermediate paths through the code.DFT tends to reflect dependencies but it is mainly through sequences of data manipulation. In short, each data variable is tracked and its use is verified. This approach tends to uncover bugs like variables used but not initialize, or declared but not used, and so on.

**7. Path Testing:** Path testing is where all possible paths through the code are defined and covered. It’s a time-consuming task.

**8. Loop Testing:** These strategies relate to testing single loops, concatenated loops, and nested loops. Independent and dependent code loops and values are tested by this approach.

## Why we perform WBT?

**To ensure:**

- That all independent paths within a module have been exercised at least once.

- All logical decisions verified on their true and false values.
- All loops executed at their boundaries and within their operational bounds internal data structures validity.

**To discover the following types of bugs:**

- Logical error tend to creep into our work when we design and implement functions, conditions or controls that are out of the program
- The design errors due to difference between logical flow of the program and the actual implementation
- Typographical errors and syntax checking

### Does this testing requires detailed programming skills?

We need to write test cases that ensure the complete coverage of the program logic.

For this we need to know the program well i.e. We should know the specification and the code to be tested. Knowledge of programming languages and logic is required for this type of testing.

### Limitations

Not possible for testing each and every path of the loops in the program. This means exhaustive testing is impossible for large systems.

This does not mean that WBT is not effective. By selecting important logical paths and data structure for testing is practically possible and effective.

## Difference Between White-Box and Black-Box Testing

To put it in simple terms:

Under Black box testing, we test the software from a user's point of view, but in White box, we see and test the actual code.

In Black box testing, we perform testing without seeing the internal system code, but in WBT we do see and test the internal code.

White box testing technique is used by both the developers as well as testers. It helps them to understand which line of code is actually executed and which is not. This may indicate that there is either a missing logic or a typo, which eventually can lead to some negative consequences.

**Recommended read => A complete guide to Black Box testing**

## Steps to Perform WBT

**Step #1** – Understand the functionality of an application through its source code. Which means that a tester must be well versed with the programming language and the other tools as well techniques used to develop the software.

**Step #2**– Create the tests and execute them.

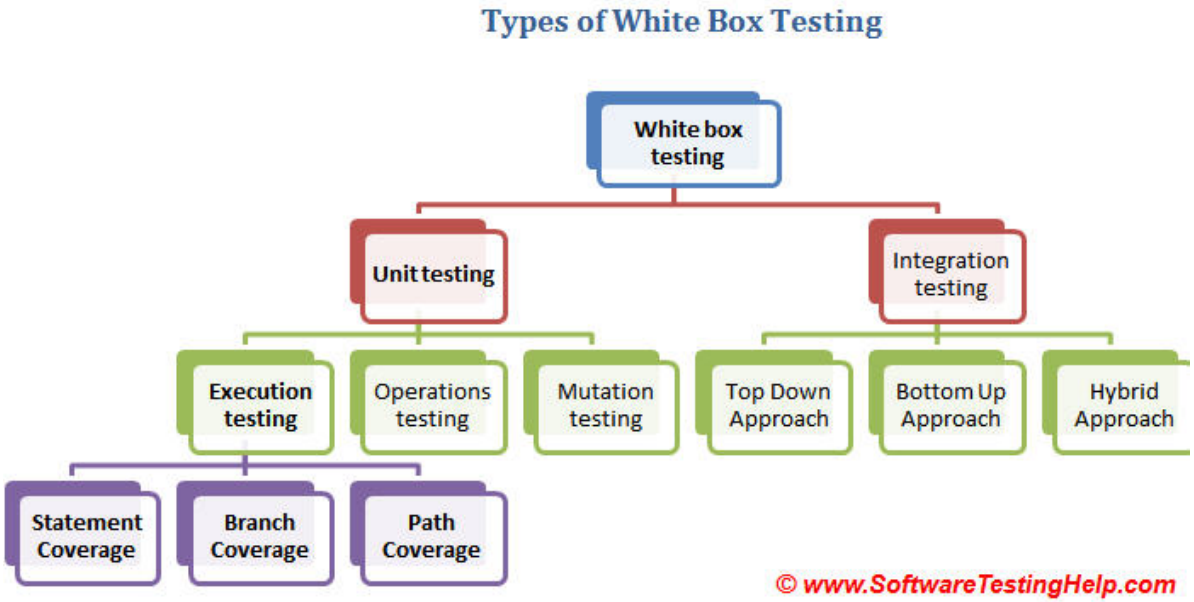
When we discuss the concept of testing, "coverage" is considered to be the most important factor. Here I will explain how to have maximum coverage from the context of White box testing.

**Also read =>** Cause and Effect Graph – Dynamic Test Case Writing Technique For Maximum Coverage

## Types and Techniques of White Box Testing

There are several types and different methods for each white box testing type.

See the below image for your reference.




*Today, we are going to focus mainly on the* ***execution testing types of 'Unit testing white box technique'.***

**3 Main White Box Testing Techniques:**

1. Statement Coverage
2. Branch Coverage
3. Path Coverage

Note that the statement, branch or path coverage does not identify any bug or defect that needs to be fixed. It only identifies those lines of code which are either never executed or remains untouched. Based on this further testing can be focused on.

Let's understand these techniques one by one with a simple example.

**#1) Statement coverage:**

In a programming language, a statement is nothing but the line of code or instruction for the computer to understand and act accordingly. A statement becomes an executable statement when it gets compiled and converted into the object code and performs the action when the program is in a running mode.

Hence *"Statement Coverage"*, as the name itself suggests, it is the method of validating whether each and every line of the code is executed at least once.

### #2) Branch Coverage:

"Branch" in a programming language is like the "IF statements". An IF statement has two branches: T**rue and False**.

So in Branch coverage (also called Decision coverage), we validate whether each branch is executed at least once.

**In case of an "IF statement", there will be two test conditions:**

- One to validate the true branch and,
- Other to validate the false branch.

Hence, in theory, Branch Coverage is a testing method which is when executed ensures that each and every branch from each decision point is executed.

### #3) Path Coverage

Path coverage tests all the paths of the program. This is a comprehensive technique which ensures that all the paths of the program are traversed at least once. Path Coverage is even more powerful than Branch coverage. This technique is useful for testing the complex programs.

***Let's take a simple example to understand all these white box testing techniques.***

**Also check => Different Types of testing**

## White Box Testing Example

***Consider the below simple pseudocode:***

```
INPUT A & B
C = A + B
```

```
IF C>100
PRINT “ITS DONE”
```

For ***Statement Coverage*** – we would only need one test case to check all the lines of the code.

**That means:**

If I consider *TestCase_01 to be (A=40 and B=70),* then all the lines of code will be executed.

**Now the question arises:**

1. Is that sufficient?
2. What if I consider my Test case as A=33 and B=45?
   Because Statement coverage will only cover the true side, for the pseudo code, only one test case would NOT be sufficient to test it. As a tester, we have to consider the negative cases as well.

   Hence for maximum coverage, we need to consider **“*Branch Coverage*”**, which will evaluate the “FALSE” conditions.

   In the real world, you may add appropriate statements when the condition fails.

***So now the pseudocode becomes:***

```
INPUT A & B

C = A + B
IF C>100
PRINT “ITS DONE”
ELSE

PRINT “ITS PENDING”
```

Since Statement coverage is not sufficient to test the entire pseudo code, we would require Branch coverage to ensure maximum coverage**.**

So for Branch coverage, we would require two test cases to complete the testing of this pseudo code.

**TestCase_01**: A=33, B=45

**TestCase_02**: A=25, B=30

With this, we can see that each and every line of the code is executed at least once.

**Here are the Conclusions that are derived so far:**



- Branch Coverage ensures more coverage than Statement coverage.
- Branch coverage is more powerful than Statement coverage.
- 100% Branch coverage itself means 100% statement coverage.
- But 100 % statement coverage does not guarantee 100% branch coverage.

Now let's move on to ***Path Coverage:***

As said earlier, Path coverage is used to test the complex code snippets, which basically involve loop statements or combination of loops and decision statements.

***Consider this pseudocode:***

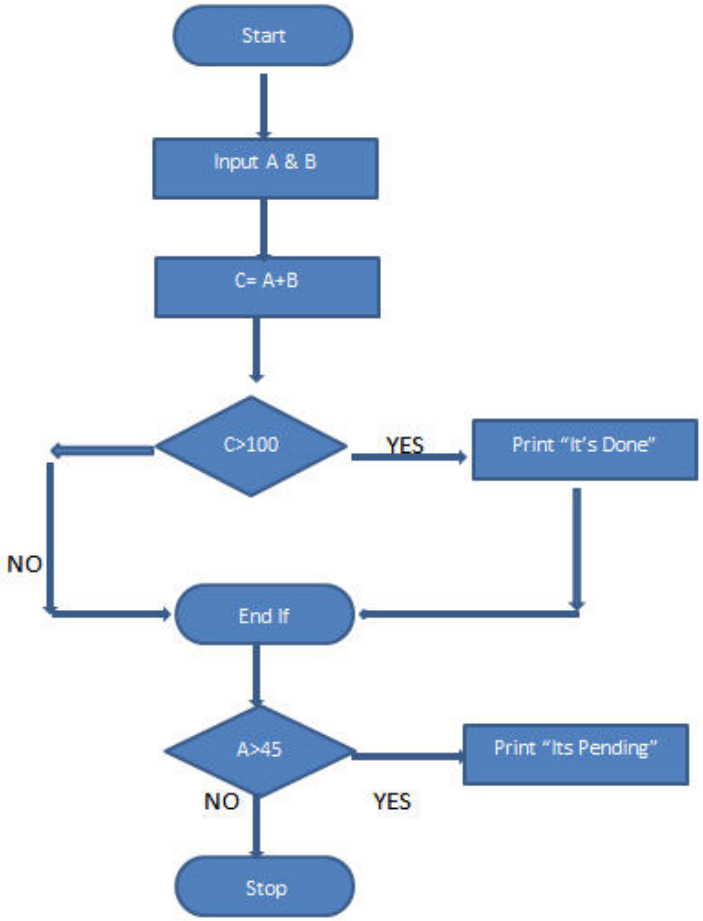
```
INPUT A & B
C = A + B
IF C>100
PRINT “ITS DONE”

END IF
IF A>50
PRINT “ITS PENDING”
END IF
```

Now to ensure maximum coverage, we would require 4 test cases.

How? Simply – there are 2 decision statements, so for each decision statement, we would need two branches to test. One for true and the other for the false condition. So for 2 decision statements, we would require 2 test cases to test the true side and 2 test cases to test the false side, which makes a total of 4 test cases.

**To simplify these let’s consider below flowchart of the pseudo code we have:**




**Path Coverage**

**In order to have the full coverage, we would need following test cases:**

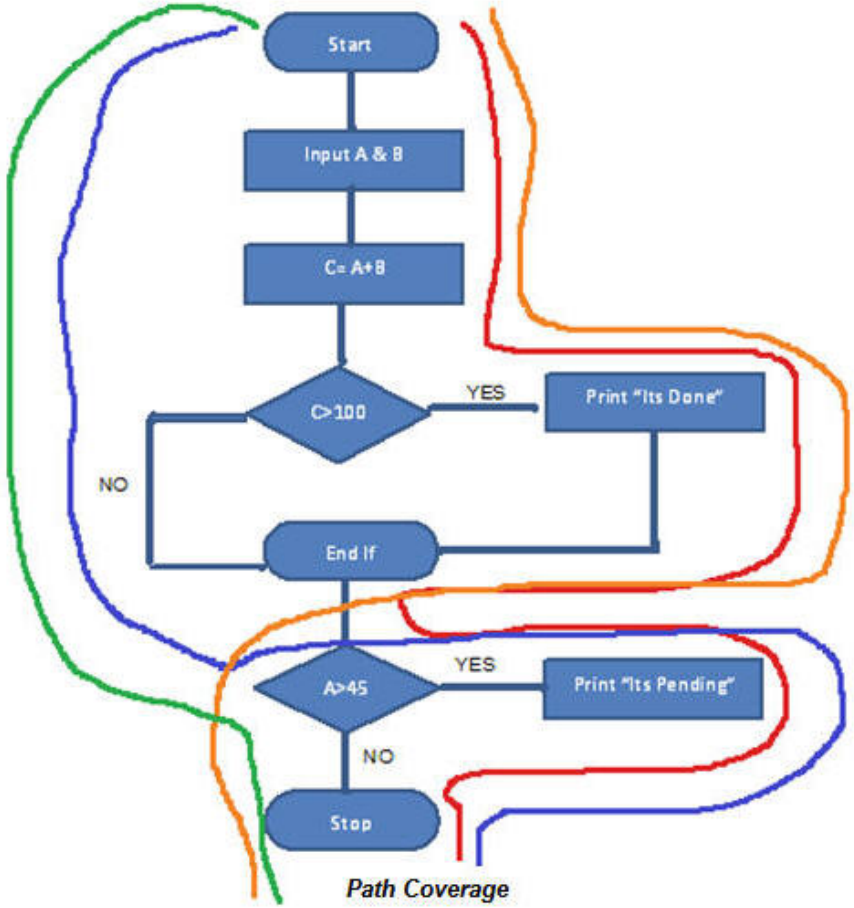
**TestCase_01:** A=50, B=60

**TestCase_02**: A=55, B=40

**TestCase_03:** A=40, B=65

**TestCase_04:** A=30, B=30

**So the path covered will be:**




Red Line – TestCase_01 = (A=50, B=60)

Blue Line = TestCase_02 = (A=55, B=40)

Orange Line = TestCase_03 = (A=40, B=65)

Green Line = TestCase_04 = (A=30, B=30)

**********

=>> **Contact us** to suggest your listing here

**********

## White Box Testing Tools

Given below is a list of top white box test tools.

### #1) Veracode


Veracode's white box testing tools will help you in identifying and resolving the software flaws quickly and easily at a reduced cost. It supports several application languages like .NET, C++, JAVA etc and also enables you to test the security of desktop, web as well as mobile applications. Still, there are several other benefits of Veracode tool. For detailed information about Veracode White box test tools, please check the below link.

Website Link: Veracode

### #2) EclEmma


EclEmma was initially designed for test runs and analysis within the Eclipse workbench. It is considered to be a free Java code coverage tool and has several features as well. To install or know more about EclEmma please check out the below link.

Website Link: EclEmma

### #3)RCUNIT

A framework which is used for testing C programs is known as RCUNIT. RCUNIT can be used accordingly based on the terms of the MIT License. It is free to use and in order to install or know more about it, please check the below link.

Website Link: RCUNIT

**#4) cfix**

cfix is one of the unit testing frameworks for C/C++ which solely aims at making test suites development as simple and easy as possible. Meanwhile, cfix is typically specialized for NT Kernel mode and Win32. To install and know more about cfix, please check out the below link

Website Link: cfix

**#5) Googletest**


Googletest is Google's C++ test framework. Test Discovery, Death tests, Value-parameterized tests, fatal & non-fatal failures, XML test report generation etc are few features of GoogleTest but there are several other features too. Linux, Windows, Symbian, Mac OS X are few platforms where GoogleTest has been used. In order to Download, please check the below link.

Download Link: Googletest

**#6) EMMA**

Emma is an easy to use free JAVA code coverage tool. It includes several features and benefits. To Download and know more about Emma, please check the below link.

Download Link: EMMA

**#7) NUnit**


NUnit is an easy to use open source unit testing framework which does not require any manual intervention to judge the test results. It supports all .NET languages. It also supports data-driven tests and tests run parallel under NUnit. Earlier releases of NUnit used NUnit license but NUnit 3 is released under the MIT license. But both the licenses allow free use without any restrictions. In order to download and know more about NUnit please check the below link.

Download Link: NUnit

**#8) CppUnit**


CppUnit is a unit testing framework written in C++ and is considered to be the port of JUnit. The test output for CppUnit may be either in the XML or text format. It creates unit tests with its own class and runs tests in the test suites. It is licensed under LGPL. In order to download and know more about CppUnit please check the below link.

Download Link: CppUnit

**#9) JUnit**


JUnit is a quiet simple unit testing framework that supports test automation in Java Programming Language. It mainly supports in Test Driven Development and provides the Test coverage report as well. It is licensed under Eclipse Public License. For free download and in order to know more about JUnit please check the below link.

Download Link: JUnit

**#10) JsUnit**


JsUnit is considered to be the port of JUnit to javascript. And it is an open source unit testing framework to support Client sided Javascript. It is licensed under GNU Public License 2.0, GNU Lesser Public License 2.1 and Mozilla Public License 1.1. In order to download and know more about JsUnit please check the below link.

Download Link: JsUnit

*Also, check all the tools that we have listed under **Static code analysis here**.*

***Feel free to suggest more simple or advanced tools that you are using for white box technique.***

## Conclusion

Relying only on black box testing is not sufficient for maximum test coverage. We need to have a combination of both black box and white box testing techniques to cover maximum defects.

If done properly, White box testing will certainly contribute to the software quality. It's also good for testers to participate in this testing as it can provide the most "unbiased" opinion about the code. :)

***Let us know if you have any questions about the methods we discussed in this article.***



← **PREV POST** | **NEXT POST** →



**45 comments ↓**

#1 **vismay**

Hello Sir,

I am vismay.

I am working as a QA Engineer in company and have 3 years of experience in manual testing and 6 month experience in automation.

I want to start freelancing please give me suggestions regarding this, how can i start and prerequisites?

#2 **Nikolay**
Good article. A very thorough discussion with nice examples.

#3 **komal**
hi
the example shared is very simple and easy to understand clearing all concepts.

thnak you

#4 **Ranjan**
Very nice.
In the Branch Coverage, the example is incorrect. Psl correct.

TestCase_01: A=33, B=45

TestCase_02: A=25, B=30

In this case, it allways goes to else part, never executed first part.

#5 **Kunjal Gandhi**
Good one.. Today I get cleared of concept behind 100% branch and statement coverage

#6 **Sumit**
can you provide white box testing tools?

#7 **Chetan Metker**
hello shilpa mem & Vijay sir:This is a good article about how to perform WBT.actually most of the people follow the BBT because they have not proper knowledge about coding,tools or frameworks etc.but in thie time in market WBT is most important technique to find a defect in software application or product.
I hope this article gives a new direction to think about WBT.
Thanks for this article…!!

#8 **gourav**
Can you please explain Path coverage in more details?

#9 **hritik**

can you provide flex testing tools,please note down here

#10 **Anand Awate**
In the Branch Coverage, the example is incorrect. Psl correct.

TestCase_01: A=33, B=45

TestCase_02: A=25, B=30

In this case, it allways goes to else part, never executed first part.

#11 **Andy**
A clearer example of when full path should be used would be helpful. Full path can be extremely time consuming, and so long as full decision and statement coverage has been achieved, it might also be unnecessary. Typically full path is reserved for safety or critical systems.

#12 **Kelis**
A really good answer, full of raotntaliiy!

#13 **Teerna**
I read so many stuffs on White Box testing but didn't understand until I read this article. Thanks! Thorough and easy to understand!

#14 **D.Selvakumar**
How i can learn this, conducting any course this in Chennai or through online.

#15 **nithhya**
hi mam nd sir ,ur articleis good nd in my opinion video explanation vil b dam good than this plz upload video explanation
thank u

#16 **P Prasanti lata**
I am a beginner in testing , really very nice aricle

#17 **P Prasanti lata**
Can any one please explain the completely the difference between condition coverage & path coverage or they are same?

#18 **etaert**
efteg

#19 **Ganesh Wagh**
Good article very helpful

#20 **karthick kumar**
very nice to understand! thank u

#21 **Leela**
Wow …much useful with clear explanation .. Tank u sooo muchhh

#22 **Raza Ali**
Its, really greate explanation

#23 **GEETHA**
GOOD

#24 **reva**
give me some example for testing

#25 **Swapnesh**
Nice

#26 **shilpa fate**
very nice explaination with example.

#27 **Anantha**
Wonderful ! I never thought it could be explained so nice and in such way that it is easy to understand.

#28 **ishan**
Good article very helpful and
very nice explaination with example.

#29 **Ms K.**
Very Good Explanation. Many thanks for posting it. I am currently preparing for CTFL . Please suggest me any good reads as my exam is due in 2 and half weeks time. I am an ICT teacher want to switch in software testing.

#30 **ssp**
give some s/w development eg….

#31 **Yasmin**
I can’t understand with full path coverage concept, can u please anyone can explain.

#32 **hiral**

hii
i have 1 year experience in manual teasting. i shift in pune so please suggestion me how to got a job in pune??

#33 **reshma**
This is very good artical.

#34 **ramya**
can i have about white box testing and its types with any simple C language code as an example and how the code is analysed in each test

#35 **IIDUMBASSII**
help me how to use whitebox pm in my email t.y

#36 **rohit shingade**
Hi vismay i am rohit plzz send your details we provide online trainings for our students all over the globe if our client needs such technology we wilp contact you.
Rohit Shingade
Training Director
StarTech online Training

#37 **Allen Urban**
It’s an interesting article, but GenRocket has an automated platform and process that does white box testing, fully tests and cuts the time and cost of testng way down.

#38 **Manzur**
Thanks for the nicely organized article.

I have a confusion about the ‘path coverage’. Test case 1 and Test Case 3 (in fact Test case 2 as well) is a repetitive test for the TRUE side of decision statement 2. Here the total independent path is identified 4. But if we apply the cyclomatic complexity metric here, the total number of independent path would be 3; which reduces the requirement of 1 test case.

#39 **Nuwangi**
Nicely explained. Very clear, Thanks

#40 **Aleksandar Ereman**
where can i read about operations testing or all if the unit testing

#41 **dipak**
I sir mam. I have just complete software testing course and i want a job . atleast i want experience salary does’nt matter.

#42 **Yevhenii**
Can you please help and draw flowchart for this task.

1. Pick up and read the newspaper.
2. Look at what is on television.
3. If there is a program that you are interested in watching then switch the television on and watch the program.
4. Otherwise.
5. Continue reading the newspaper.
6. If there is a crossword in the newspaper then try and complete the crossword.

A. SC = 1 and DC = 3.
B. SC = 1 and DC = 2.
C. SC = 2 and DC = 3.
D. SC = 2 and DC = 2.
E. SC = 1 and DC = 1.

#43 **Chris**
Thank you finally got it <3

#44 **khaleeq**
can you please discuss about insertion sort test using white box approach.

#45 **Rahul Baniya**
Can you group with me for BE project.
i love sassy and her assy.

## Leave a Comment

**Name***

**Email***

**Website (Optional)**

Submit

Search





Thanks For Reading! 50 Million Visits and Counting!