Typical black-box test design techniques include:

- Equivalence partitioning.
- Boundary value analysis.
- Decision table testing.
- All-pairs testing.
- Cause–effect graph.
- Error guessing
- State transition

- **Equivalence partitioning:**
  The idea behind this technique is to divide (i.e. to partition) a set of test conditions into groups or sets that can be considered the same (i.e. the system should handle them equivalently), hence 'equivalence partitioning or we can say set of valid and invalid classes

```
Assume that the application accepts an integer in the range 100 to 999

Valid Equivalence Class partition: 100 to 999 inclusive.

Non-valid Equivalence Class partitions: less than 100, more than 999, decimal numbers and
alphabets/non-numeric characters.
```

Rule: <min, min-max, >max

A text field permits only numeric characters Length must be 6-10 characters long

< 6 , 6-10, >6

What is Boundary value analysis:

Boundary value analysis is a test case design technique to test boundary value between partitions (both valid boundary partition and invalid boundary partition). A boundary value is an input or output value on the border of an equivalence partition, includes minimum and maximum values at inside and outside boundaries.

Rule: Min-1, min, min+1, max-1, max, max+1

If A text field permits only numeric characters Length must be 6-10 characters long

5,6,7,9,10,11

1. What is an equivalence partition (also known as an equivalence class)?

a) A set of test cases for testing classes of objects.

b) An input or output range of values such that only one value in the range becomes a test case.

c) An input or output range of values such that each value in the range becomes a test case.

d) An input or output range of values such that every tenth value in the range becomes a test case.

Ans: B

Question 1

One of the fields on a form contains a text box which accepts numeric values in the range of 18 to 25. Identify the invalid Equivalence class.

a) 17
b) 19
c) 24
d) 21

Solution

The text box accepts numeric values in the range 18 to 25 (18 and 25 are also part of the class). So this class becomes our valid class. But the question is to identify invalid equivalence class. The classes will be as follows:

Class I: values < 18 => invalid class
Class II: 18 to 25 => valid class
Class III: values > 25 => invalid class

17 fall under invalid class. 19, 24 and 21 fall under valid class. **So answer is 'A'**

Question 2

In an Examination a candidate has to score minimum of 24 marks in order to clear the exam. The maximum that he can score is 40 marks. Identify the Valid Equivalence values if the student clears the exam.

a) 22,23,26
b) 21,39,40
c) 29,30,31
d) 0,15,22

Solution

The classes will be as follows:

Class I: values < 24 => invalid class
Class II: 24 to 40 => valid class
Class III: values > 40 => invalid class

We have to indentify Valid Equivalence values. Valid Equivalence values will be there in Valid Equivalence class. All the values should be in Class II. **So answer is 'C'**

Question 3

One of the fields on a form contains a text box which accepts alpha numeric values. Identify the Valid Equivalence class

a) BOOK
b) Book
c) Boo01k
d) Book

Solution

Alpha numeric is combination of alphabets and numbers. Hence we have to choose an

option which has both of these. A valid equivalence class will consist of both alphabets and numbers. Option 'c' contains both alphabets and numbers. **So answer is 'C'**

Question 4

The Switch is switched off once the temperature falls below 18 and then it is turned on when the temperature is more than 21. When the temperature is more than 21. Identify the Equivalence values which belong to the same class.

a) 12,16,22
b) 24,27,17
c) 22,23,24
d) 14,15,19

Solution

We have to choose values from same class (it can be valid or invalid class). The classes will be as follows:
Class I: less than 18 (switch turned off)
Class II: 18 to 21
Class III: above 21 (switch turned on)
Only in Option 'c' all values are from one class. Hence the **answer is 'C'**. (Please note that the question does not talk about valid or invalid classes. It is only about values in same class)

Question 5

A program validates a numeric field as follows: values less than 10 are rejected, values between 10 and 21 are accepted, values greater than or equal to 22 are rejected. Which of the following input values cover all of the equivalence partitions?

a. 10,11,21
b. 3,20,21
c. 3,10,22
d. 10,21,22

Solution

We have to select values which fall in all the equivalence class (valid and invalid both). The classes will be as follows:
Class I: values <= 9 => invalid class
Class II: 10 to 21 => valid class
Class III: values >= 22 => invalid class
All the values from option 'c' fall under all different equivalence class. **So answer is 'C'.**

7. The specification: an integer field shall contain values from and including 1 to and including 12 (number of the month). Which equivalence class partitioning is correct?

a) Less than 1, 1 through 12, larger than 12
b) Less than 1, 1 through 11, larger than 12
c) Less than 0, 1 through 12, larger than 12
d) Less than 1, 1 through 11, and above
Ans: a

Question 6

A program validates a numeric field as follows: values less than 10 are rejected, values

between 10 and 21 are accepted, values greater than or equal to 22 are rejected. Which of the following covers the MOST boundary values?
a. 9,10,11,22
b. 9,10,21,22
c. 10,11,21,22
d. 10,11,20,21

Solution

We have already come up with the classes as shown in question 5. The boundaries can be identified as 9, 10, 21, and 22. These four values are in option 'b'. **So answer is 'B'**

Question 7

In a system designed to work out the tax to be paid:
An employee has £4000 of salary tax free.
The next £1500 is taxed at 10%.
The next £28000 after that is taxed at 22%.
Any further amount is taxed at 40%.
To the nearest whole pound, which of these groups of numbers fall into three DIFFERENT equivalence classes?
a) £4000; £5000; £5500
b) £32001; £34000; £36500
c) £28000; £28001; £32001
d) £4000; £4200; £5600

Solution

The classes will be as follows:
Class I : 0 to £4000 => no tax
Class II : £4001 to £5500 => 10 % tax
Class III : £5501 to £33500 => 22 % tax
Class IV : £33501 and above => 40 % tax
Select the values which fall in three different equivalence classes. Option 'd' has values from three different equivalence classes. **So answer is 'D'.**

Question 8

In a system designed to work out the tax to be paid:
An employee has £4000 of salary tax free.
The next £1500 is taxed at 10%.
The next £28000 after that is taxed at 22%.
Any further amount is taxed at 40%.
To the nearest whole pound, which of these is a valid Boundary Value Analysis test case?
a) £28000
b) £33501
c) £32001
d) £1500

Solution

The classes are already divided in question # 7. We have to select a value which is a boundary value (start/end value). 33501 is a boundary value. **So answer is 'B'**.
10. Order numbers on a stock control system can range between 10000 and 99999 inclusive. Which of the following inputs might be a result of designing tests for only valid equivalence classes and valid boundaries:

a) 1000, 5000, 99999
b) 9999, 50000, 100000
c) 10000, 50000, 99999
d) 10000, 99999
e) 9999, 10000, 50000, 99999, 10000
Ans: C
11. Equivalence partitioning is:

a. A black box testing technique used only by developers
b. A black box testing technique than can only be used during system testing
c. A black box testing technique appropriate to all levels of testing
d. A white box testing technique appropriate for component testing
Ans: C

12. Which of the following is a valid collection of equivalence classes for the following problem: An integer field shall contain values from and including 1 to and including 15

a. Less than 1, 1 through 15, more than 15
b. Negative numbers, 1 through 15, above 15
c. Less than 1, 1 through 14, more than 15
d. Less than 0, 1 through 14, 15 and more

Ans: a

13. Which of the following is a valid collection of equivalence classes for the following problem: Paying with credit cards shall be possible with Visa, Master and Amex cards only.

a. Visa, Master, Amex;
b. Visa, Master, Amex, Diners, Keycards, and other option
c. Visa, Master, Amex, any other card, no card
d. No card, other cards, any of Visa – Master – Amex

Ans: d

15. One of the fields on a form contains a text box which accepts alphabets in lower or upper case.Indentify the invalid Equivalence class value.

a. CLASS
b. cLASS
c. CLass
d. CLa01ss
And: D

Question 9

Given the following specification, which of the following values for age are in the SAME equivalence partition?

If you are less than 18, you are too young to be insured.

Between 18 and 30 inclusive, you will receive a 20% discount.

Anyone over 30 is not eligible for a discount.

a) 17, 18, 19

b) 29, 30, 31

c) 18, 29, 30

d) 17, 29, 31

Solution

The classes will be as follows:

Class I: age < 18 => not insured

Class II: age 18 to 30 => 20 % discount

Class III: age > 30 => no discount

Here we cannot determine if the above classes are valid or invalid, as nothing is mentioned in the question. (But according to our guess we can say I and II are valid and III is invalid. But this is not required here.) We have to select values which are in SAME equivalence partition. Values from option 'c' fall in same partition. **So answer is 'C'.**

Company ABC is going to provide their employees with a bonus which will be based on the employee's
length of service in the company. The bonus calculation will be zero if they have been with the company
for less than two years, 10% of their salary for more than two but less than five years, and 25% for five to
ten years, 35% for ten years or more. The interface will not allow a negative value to be input, but it will
allow a zero to be input.
How many equivalence partitions are needed to test the calculation of the bonus?

A. Two equivalence partitions.
B. Three equivalence partitions.
C. Four equivalence partitions.

D. Five equivalence partitions.

Ans: C

Q. 901: An employee's bonus is to be calculated. It cannot become negative, but it can be calculated to zero. The bonus is based on the duration of the employment. An employee can be employed for less than or equal to 2 years, more than 2 years but less than 5 years,

5 to 10 years, or longer than 10 years. Depending on this period of employment, an employee will get either onus or a bonus of 10%, 25% or 35%.

How many equivalence partitions are needed to test the calculation of the onus?

A. 3.

B. 5.

C. 2.

D. 4.

Ans: D

What are Decision Tables?

Decision tables are precise and compact way to model complicated logic. They are ideal for describing situations in which a number of combinations of actions are taken under varying sets of conditions.

What is the expected result for each of the following test cases?

A. Citibank card member, holding a Silver room
B. Non Citibank-member, holding a Platinum room

a) A – Don't offer any upgrade, B – Don't offer any upgrade.
b) A – Don't offer any upgrade, B – Offer upgrade to Gold.
c) A – Offer upgrade to Silver, B – Offer upgrade to Silver.
d) A – Offer upgrade to Gold, B – Don't offer any upgrade.

	Rule 1	Rule 2	Rule 3	Rule 4
Conditions				
Citibank Card Member	Yes	Yes	No	No
Type of Room	Silver	Platinum	Silver	Platinum
Actions				
Offer upgrade to Gold Luxury	Yes	No	No	No
Offer upgrade to Silver	N/A	Yes	N/A	No

Ans: D

Given the following decision table: Which of the following test cases and expected results is VALID?

	Rule 1	Rule 2	Rule 3	Rule 4
Conditions				
Age	<21 yrs	21-29 yrs	30-50yrs	> 50yrs
Insurance Class	A	A or B	B. C or D	C or D
Actions				
Premium	£100	£90	£70	£70
Excess	£2,500	£2,500	£500	£1000

A.23 year old in insurance class A Premium is 0 and excess is,500.
B. 51 year old in insurance class C Premium is 0 and excess is 00.
C. 31 year old in insurance class B Premium is 0 and excess is ,500.
D. 43 year old in insurance class C Premium is 0 and excess is ,000.

Ans: A

Q. 1:Which of the following would be an example of decision-table testing for a financial application applied at the system-test level?

A. A table containing rules for combinations of inputs to two fields on a screen.

B. A table containing rules for interfaces between components.

C. A table containing rules for mortgage applications.

D. A table containing rules for chess.

Ans:C

Q.2: Which of the following could be a coverage measure for state transition testing?

V. All states have been reached.

W. The response time for each transaction is adequate.

X. Every transition has been exercised.

Y. All boundaries have been exercised.

Z. Specific sequences of transitions have been exercised.

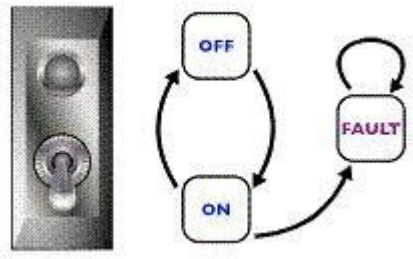
A. X, Y and Z

B. V, X, Y and Z

C. W,X and Y

Ans:D

Q Which state is invalid state transition



a) Off to On
b) On to Off
c) Fault to On

Ans: C

Cause Effect-Graph: Graphical representation of Decision table is k/a cause effect graph. This was invented by 1973 by Bill on IBM.

What is Pairwise Testing?

Pairwise Testing also known as All-pairs testing is a testing approach taken for testing the software using combinatorial method. It's a method to test all the possible discrete combinations of the parameters involved.

Assume we have a piece of software to be tested which has got 10 input fields and 10 possible settings for each input field, then there are 10^10 possible inputs to be tested. In this case, exhaustive testing in impossible even if we wish to test all combinations.

Let us also understand the concept by understanding with an example:

Example:

An application with simple list box with 10 elements (Let's say 0,1,2,3,4,5,6,7,8,9) along with a checkbox, radio button, Text Box and OK Button. The Constraint for the Text box is it can accept values only between 1 and 100. Below are the values that each one of the GUI objects can take :

List Box - 0,1,2,3,4,5,6,7,8,9

Check Box - Checked or Unchecked

Radio Button - ON or OFF

Text Box - Any Value between 1 and 100

Exhaustive combination of the product B is calculated.

```
List Box = 10

Check Box = 2

Radio Button = 2

Text Box = 100


Total Number of Test Cases using Cartesian Method : 10*2*2*100 = 4000
```

```
Total Number of Test Cases including Negative Cases will be > 4000
```

Now, the idea is to bring down the number of test cases. We will first try to find out the number of cases using the conventional software testing technique. We can consider the list box values as 0 and others as 0 is neither positive nor negative. Radio button and check box values cannot be reduced, so each one of them will have 2 combinations (ON or OFF). The Text box values can be reduced into three inputs (Valid Integer, Invalid Integer, Alpha-Special Character).

Now, we will calculate the number of cases using software testing technique, 2*2*2*3 = 24 (including negative cases).

Now, we can still reduce the combination further into All-pairs technique.

Step 1: Order the values such that one with most number of values is the first and the least is placed as the last variable.

Step 2: Now start filling the table column by column. List box can take 2 values.

Step 3: The Next column under discussion would be check box. Again Check box can take 2 values.

Step 4: Now we need to ensure that we cover all combinations between list box and Check box.

Step 5: Now we will use the same strategy for checking the Radio Button. It can take 2 values.

Step 6: Verify if all the pair values are covered as shown in the table below.

Text Box	List Box	Check Box	Radio Button
Valid Int	0	check	ON
Valid Int	others	uncheck	OFF
Invalid Int	0	check	ON
Invalid Int	others	uncheck	OFF
AlphaSpecialCharacter	0	check	ON

AlphaSpecialCharacter	others	uncheck	OFF

Error Guessing: comes with experience with the technology and the project. Error Guessing is the art of guessing where errors can be hidden. Like Banking BUG of Amazon and Flipkart.

- **State transition:** State Transition testing, a black box testing technique, in which outputs are triggered by changes to the input conditions or changes to 'state' of the system. In other words, tests are designed to execute valid and invalid state transitions.
  When to use? When we have sequence of events that occur and associated conditions that apply to those events.
  Like ATM Example

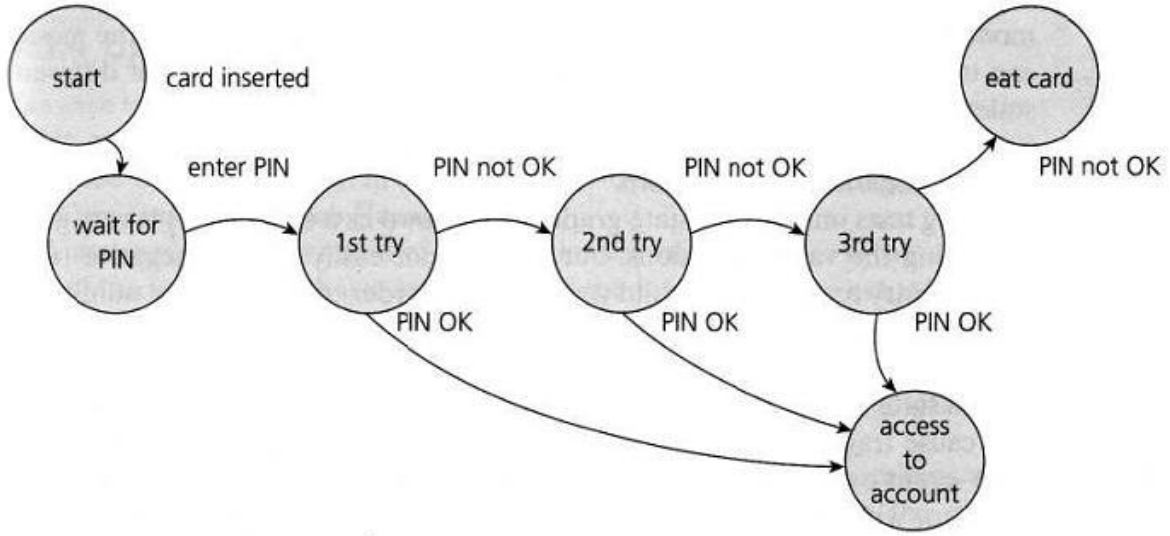
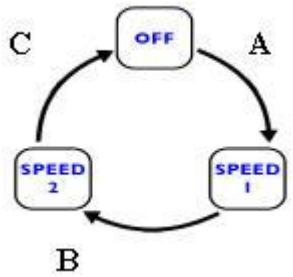


FIGURE 4.2 State diagram for PIN entry

In deriving test cases, we may start with a typical scenario.

- First test case here would be the normal situation, where the correct PIN is entered the first time.
- A second test (to visit every state) would be to enter an incorrect PIN each time, so that the system eats the card.
- A third test we can do where the PIN was incorrect the first time but OK the second time, and another test where the PIN was correct on the third try. These tests are probably less important than the first two.
- Note that a transition does not need to change to a different state (although all of the transitions shown above do go to a different state). So there could be a transition from ‘access account’ which just goes back to ‘access account’ for an action such as ‘request balance’.
-

Consider the following state transition diagram of a two-speed hair dryer, which is operated by pressing its one button. The first press of the button turns it on to Speed 1, second press to Speed 2 and the third press turns it off.



Which of the following series of state transitions below will provide 0-switch coverage?

a. A,C,B

b. B,C,A

c. A,B,C

d. C,B,A

Evaluating the options:

In State transition testing a test is defined for each state transition. The coverage that is achieved by this testing is called 0-switch or branch coverage. 0-switch coverage is to execute each loop once (No repetition. We should start with initial state and go till end state. It does not test 'sequence of two state transitions'). In this case the start state is 'OFF', and then press of the button turns it on to Speed 1 (i.e. A). Second press turns it on to Speed 2 (i.e. B) and the third press turns it off (i.e. C). Here we do not test the combinations like what if the start state is 'Speed 1' or 'Speed 2' etc.

An alternate way of solving this is check for the options where it starts with 'OFF' state. So we have options 'a' and 'c' to select from. As per the state diagram from 'OFF' state the dryer goes to 'Speed 1' and then to 'Speed 2'. So our answer should start with 'A' and end with 'C'.

The answer is 'C'