



**East West Institute of Technology**  
**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**  
**(Affiliated to Visvesvaraya Technological University, Belagavi)**  
Bengaluru-91



# **SOFTWARE TESTING LABORATORY**

## **(21ISL66)**

Academic year: 2022-2023  
Semester: VI

Prepared By,  
Ms.Pratyaksha S, Mrs Sharon  
Assistant Professor  
Department of ISE  
EWIT, Bangalore

Verified By  
Dr.Suresh M B  
Professor & HOD  
Department of ISE  
EWIT,Bangalore

## **SOFTWARE TESTING LABORATORY**

Subject Code: 21ISL66

Hours/Week: 03

Number of Contact Hours/Week 0:2:2

Total Number of Lab Contact Hours: 36

I.AMarks:50

Exam Hours: 3

Exam Marks: 100

1. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of boundary value testing, derive different test cases, execute these test cases and discuss the test results.
2. Design, develop, code and run the program in any suitable language to implement the NextDate function. Analyze it from the perspective of equivalence class value testing, derive different test cases, execute these test cases and discuss the test results.
3. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of decision table-based testing, derive different test cases, execute these test cases and discuss the test results.
4. Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on boundary-value analysis, equivalence class partitioning and decision-table approach and execute the test cases and discuss the results
5. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of dataflow testing, derive different test cases, execute these test cases and discuss the test results.
6. Design, develop, code and run the program in any suitable language to implement the binary search algorithm. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results.

### **PART-B PRATICAL BASED LEARNING**

Develop a Mini Project with documentation of suitable test-cases and their results to perform automation testing of any E-commerce or social media web page

1. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of boundary value testing, derive different test cases, execute these test cases and discuss the test results.

### ALGORITHM

STEP 1: Define lockPrice=45.0, stockPrice=30.0, barrelPrice=25.0

STEP2: Input locks

STEP3: while(locks!=-1) ,,input device uses -1 to indicate end of data  
goto STEP 12

STEP4:input (stocks, barrels)

STEP5: compute lockSales, stockSales, barrelSales and sales

STEP6: output("Total sales:" sales)

STEP7: if (sales > 1800.0) goto STEP 8 else goto STEP 9

STEP8: commission=0.10\*1000.0; commission=commission+0.15 \* 800.0;  
commission = commission + 0.20 \* (sales-1800.0)

STEP9: if (sales > 1000.0) goto STEP 10 else goto STEP 11

STEP10: commission=0.10\* 1000.0; commission=commission + 0.15 \*  
(sales-1000.0)

STEP11: Output("Commission is \$", commission)

STEP12: exit

## PROGRAM CODE:

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int locks, stocks, barrels, t_sales, flag = 0;
    float commission;
    clrscr();
    printf("Enter the total number of locks");
    scanf("%d",&locks);
    if ((locks <= 0) || (locks > 70))
    {
        flag = 1;
    }
    printf("Enter the total number of stocks");
    scanf("%d",&stocks);
    if ((stocks <= 0) || (stocks > 80))
    {
        flag = 1;
    }
    printf("Enter the total number of barrelss");
    scanf("%d",&barrels);
    if ((barrels <= 0) || (barrels > 90))
    {
        flag = 1;
    }
    if (flag == 1)
    {
        printf("invalid input");
        getch();
        exit(0);
    }

    t_sales = (locks * 45) + (stocks * 30) + (barrels * 25);

    if (t_sales <= 1000)
    {
        commission = 0.10 * t_sales;
    }
}
```

```
    else if (t_sales < 1800)
    {
        commission = 0.10 * 1000;
        commission = commission + (0.15 * (t_sales - 1000));
    }
    else
    {
        commission = 0.10 * 1000;
        commission = commission + (0.15 * 800);
        commission = commission + (0.20 * (t_sales - 1800));
    }
    printf("The total sales is %d \n The commission is %f",t_sales,
    commission);
    getch();
    return;
}
```

**Test Case:**

**Test data:** price Rs for lock - 45.0 , stock - 30.0 and barrel - 25.0

Sales = total lock \* lock price + total stock \* stock price + total barrel \* barrel price

Commission: 10% up to sales Rs 1000 , 15 % of the next Rs 800 and 20 % on any sales in excess of 1800

**Pre-condition:** lock = -1 to exit and  $1 \leq \text{lock} \leq 70$  ,  $1 \leq \text{stock} \leq 80$  and  $1 \leq \text{barrel} \leq 90$

**Brief Description:** The salesperson had to sell at least one complete rifle per month

Case Id	Description	Input Data			Expected Output	Actual Output	Comments
		a	b	c			
1	Enter the min value for locks, stocks and barrels	1	1	1	100	10	
2	Enter the min value for 2 items and min +1 for any one item	1	1	2	125	12.5	
3	Enter the value sales approximately mid value between 100 to 1000	5	5	5	500	50	
4	Enter the values to calculate the commission for sales nearly greater than 1800	18	18	19	1825	225	
5	Enter the max value for 2 items and max - 1 for any one item	70	80	89	7775	1415	

2. Design, develop, code and run the program in any suitable language to implement the NextDate function. Analyze it from the perspective of boundary value testing, derive different test cases, execute these test cases and discuss the test results.

### ALGORITHM

STEP 1: Input date in format DD.MM.YYYY

STEP2: if MM is 01, 03, 05, 07, 08, 10 do STEP3 else STEP6

STEP3: if DD < 31 then do STEP4 else if DD=31 do STEP5 else  
output(Invalid Date);

STEP4: tomorrowday=DD+1 goto STEP18

STEP5: tomorrowday=1; tomorrowmonth=month + 1 goto STEP18

STEP6: if MM is 04, 06, 09, 11 do STEP7

STEP7: if DD<30 then do STEP4 else if DD=30 do STEP5 else  
output(Invalid Date);

STEP8: if MM is 12

STEP9: if DD<31 then STEP4 else STEP10

STEP10: tomorrowday=1, tomorrowmonth=1,  
tomorrowyear=YYYY+1; goto STEP18

STEP11: if MM is 2

STEP12: if DD<28 do STEP4 else do STEP13

STEP13: if DD=28 & YYYY is a leap do STEP14 else STEP15

STEP14: tomorrowday=29 goto STEP18

STEP15: tomorrowday=1, tomorrowmonth=3, goto STEP18;

STEP16: if DD=29 then do STEP15 else STEP17

STEP17: output("Cannot have feb", DD); STEP19

STEP18: output(tomorrowday, tomorrowmonth, tomorrowyear);

STEP19: exit



## PROGRAM CODE:

```
#include<stdio.h>
#include<conio.h>
main( )
{
    int
    month[12]={ 31,28,31,30,31,30,31,31,30,31,30,31};
    int d,m,y,nd,nm,ny,ndays;
    clrscr( );
    printf("enter the date,month,year");
    scanf("%d%d%d",&d,&m,&y);

    ndays=month[m-1];

    if(y<=1812 && y>2012)
    {
        printf("Invalid Input Year");
        exit(0);
    }

    if(d<=0 || d>ndays)
    {
        printf("Invalid Input Day");
        exit(0);
    }

    if(m<1 && m>12)
    {
        printf("Invalid Input Month");
        exit(0);
    }

    if(m==2)
    {
        if(y%100==0)
        {
            if(y%400==0)
                ndays=29;
        }
    }
}
```

```

        }
        else if(y%4==0)
            ndays=29;
    }

    nd=d+1;
    nm=m;
    ny=y;

    if(nd>ndays)
    {
        nd=1;
        nm++;
    }

    if(nm>12)
    {
        nm=1;
        ny++;
    }

    printf("\n Given date is %d:%d:%d",d,m,y);
    printf("\n Next day's date is %d:%d:%d",nd,nm,ny);
    getch( );
}

```

**Test Cases:****Test data :** Enter the three integer value**Pre-condition :** Month 1 to 12 , DAY 1 TO 31 AND YEAR 1812 TO 2013 / we are consider one corner of the input space

	Min	Min +1	Normal	Max -1	Max
Month	1	2	6	11	12
Day	1	2	15	30	31
Year	1812	1813	1912	2012	2013

Case Id	Description	Input Data			Expected Output		
		Month	day	year	Month	day	year
1	Enter the min value month,day and year	1	1	1812	1	2	1812
2	Enter the min+1 value for year and min for month and day	1	1	1813	1	2	1813
3	Enter the normal value for year and min for month and day	1	1	1912	1	2	1912
4	Enter the max -1 value for year and min for month and day	1	1	2012	1	2	2012
5	Enter the max value for year and min for month and day	1	1	2013	1	2	2013
6	Enter the min+1 value of day and min for month and year	1	2	1812	1	3	1812
7	Enter the min+1 value for day and year and min for month	1	2	1813	1	3	1813

3. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of decision table-based testing, derive different test cases, execute these test cases and discuss the test results.

### ALGORITHM

Step 1: Input 3 integer numbers which represents number of Locks, Stocks and Barrels sold.

Step 2: compute the total sales  $= (\text{Number of Locks sold} * 45) + (\text{Number of Stocks sold} * 30) + (\text{Number of Barrels sold} * 25)$

Step 3: if a totals sale in dollars is less than or equal to \$1000  
then commission =  $0.10 * \text{total Sales}$  do step 6

Step 4: else if total sale is less than \$1800  
then commission1 =  $0.10 * 1000$   
commission = commission1 +  $(0.15 * (\text{total sales} - 1000))$   
do step 6

Step 5: else commission1 =  $0.10 * 1000$   
commission2 = commission1 +  $(0.15 * 800)$   
commission = commission2 +  $(0.20 * (\text{total sales} - 1800))$  do  
step 6

Step 6: Print commission.

Step 7: Stop.

## PROGRAM CODE:

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int locks, stocks, barrels, t_sales, flag = 0;
    float commission;
    clrscr();
    printf("Enter the total number of locks");
    scanf("%d",&locks);
    if ((locks <= 0) || (locks > 70))
    {
        flag = 1;
    }
    printf("Enter the total number of stocks");
    scanf("%d",&stocks);
    if ((stocks <= 0) || (stocks > 80))
    {
        flag = 1;
    }
    printf("Enter the total number of barrelss");
    scanf("%d",&barrels);
    if ((barrels <= 0) || (barrels > 90))
    {
        flag = 1;
    }
    if (flag == 1)
    {
        printf("invalid input");
        getch();
        exit(0);
    }

    t_sales = (locks * 45) + (stocks * 30) + (barrels * 25);

    if (t_sales <= 1000)
    {
        commission = 0.10 * t_sales;
```

```
    }  
    else if (t_sales < 1800)  
    {  
        commission = 0.10 * 1000;  
        commission = commission + (0.15 * (t_sales - 1000));  
    }  
    else  
    {  
        commission = 0.10 * 1000;  
        commission = commission + (0.15 * 800);  
        commission = commission + (0.20 * (t_sales - 1800));  
    }  
    printf("The total sales is %d \n The commission is %f",t_sales,  
    commission);  
    getch();  
    return;  
}
```

## Test Report:

**Test data :** price Rs for lock - 45.0 , stock - 30.0 and barrel - 25.0

Sales = total lock \* lock price + total stock \* stock price + total barrel \* barrel price

Commission : 10% Upto sales Rs 1000 , 15 % of the next Rs 800 and 20 % on any sales in excess of 1800

**Pre-condition :** lock = -1 to exit and  $1 \leq \text{lock} \leq 70$  ,  $1 \leq \text{stock} \leq 80$  and  $1 \leq \text{barrel} \leq 90$

**Brief Description:** The salesperson had to sell at least one complete rifle per month.

RULES		R1	R2	R3	R4	R5	R6	R7	R8	R9
Conditions	C1:	T	F	F	F	F	F	F	F	F
	C2 :	-	T	T	F	T	F	F	F	T
	C3 :	-	T	F	T	F	T	F	F	T
	C4 :	-	F	T	T	F	F	T	F	T
Actions	a1 :	X								
	a2 :				X		X	X	X	
	a3 :				X		X		X	X
	a4 :		X			X	X		X	
	a5 :		X	X	X	X	X	X		X

4. Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on boundary-value analysis equivalence class partitioning and decision-table approach and execute the test cases and discuss the results

**ALGORITHM:**

Step 1: Input a, b & c i.e three integer values which represent three sides of the triangle.

Step 2: if  $(a < (b + c))$  and  $(b < (a + c))$  and  $(c < (a + b))$  then  
do step 3  
else  
print not a triangle. do step 6.

Step 3: if  $(a=b)$  and  $(b=c)$  then  
Print triangle formed is equilateral. do step 6.

Step 4: if  $(a \neq b)$  and  $(a \neq c)$  and  $(b \neq c)$  then  
Print triangle formed is scalene. do step 6.

Step 5: Print triangle formed is Isosceles.

Step 6: stop



## PROGRAM CODE:

```
#include<stdio.h>
#include<ctype.h>
#include<conio.h>
#include<process.h>
int main()
{
    int a, b, c;
    clrscr();

    printf("Enter three sides of the triangle");
    scanf("%d%d%d", &a, &b, &c);

    if((a > 10) || (b > 10) || (c > 10))
    {
        printf("Out of range");
        getch();
        exit(0);
    }
    if((a<b+c)&&(b<a+c)&&(c<a+b))
    {
        if((a==b)&&(b==c))
        {
            printf("Equilateral triangle");
        }
        else if((a!=b)&&(a!=c)&&(b!=c))
        {
            printf("Scalene triangle");
        }
        else
            printf("Isosceles triangle");
    }
    else
    {
        printf("triangle cannot be formed");
    }
    getch();
    return 0;
}
```

**Test Case Name: Boundary Value Analysis for triangle problem****Test Data : Enter the 3 Integer Value( a , b And c )****Pre-condition :  $1 \leq a \leq 10$  ,  $1 \leq b \leq 10$  and  $1 \leq c \leq 10$  and  $a < b + c$  ,  $b < a + c$  and  $c < a + b$** **Brief Description : Check whether given value for a equilateral, isosceles , Scalene triangle or can't form a triangle**

Case Id	Description	Input Data			Expected Output	Actual Output	Comments
		a	b	c			
1	Enter the min value for a , b and c	1	1	1	Should display the message Equilateral Triangle		
2	Enter the min value for 2 items and min +1 for any one item1	1	1	2	Message should be displayed can't form a triangle		
3	Enter the normal Value for a, b and c	5	5	5	Should display the message Equilateral triangle		
4	Enter the normal value for 2 items and 1 item is min value	1	5	5	Should display the message Isosceles triangle		

**Test Report:****Test Data : Enter the 3 Integer Value( a , b And c )****Pre-condition :  $1 \leq a \leq 10$  ,  $1 \leq b \leq 10$  and  $1 \leq c \leq 10$  and  $a < b + c$  ,  $b < a + c$  and  $c < a + b$** **Brief Description : Check whether given value for a equilateral, isosceles , Scalene triangle or can't form a triangle****Triangle Problem -Equivalence Class Test cases for input data**

Case Id	Description	Input Data			Expected Output	Actual Output	Comments
		a	b	c			
Weak Equivalence class Testing							

1	Enter the min value for a , b and c	5	5	5	Should display the message Equilateral triangle		
2	Enter the min value for a , b and c	2	2	3	Should display the message Isosceles Triangle		
3	Enter the min value for a , b and c	3	4	5	Should display the message Scalene Triangle		
<b>Weak Robust Equivalence class Testing</b>							
4	Enter one invalid input and two valid value for a , b and c	-1	5	5	Should display value of a is not in the range of permitted values		
5	Enter one invalid input and two valid value for a , b and c	5	-1	5	Should display value of a is not in the range of permitted values		
6	Enter one invalid input and two valid value for a , b and c	5	5	-1	Should display value of a is not in the range of permitted values		
<b>Strong Robust Equivalence class Testing</b>							
7	Enter two invalid	-1	-1	5	Should display value of a is		

	input and two valid value for a , b and c				not in the range of permitted values		
8	Enter two invalid input and two valid value for a , b and c	-1	5	-1	Should display value of a is not in the range of permitted values		

## Test Report:

Test Data: Enter the 3 Integer Value( a , b And c ) Pre-condition :  $a < b + c$  ,  $b < a + c$  and  $c < a + b$

Brief Description: Check whether given value for a equilateral, isosceles, Scalene triangle or can't form a triangle

Input data decision Table

RULES		R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11
Conditions	C1: $a < b + c$	F	T	T	T	T	T	T	T	T	T	T
	C2: $b < a + c$	-	F	T	T	T	T	T	T	T	T	T
	C3: $c < a + b$	-	-	F	T	T	T	T	T	T	T	T
	C4: $a = b$	-	-	-	T	T	T	T	F	F	F	F
	C5: $a = c$	-	-	-	T	T	F	F	T	T	F	F
	C6: $b = c$	-	-	-	T	F	T	F	T	F	T	F
Actions	a1	X	X									
	a2										X	
	a3 : Isosceles triangle							X		X	X	
	a4			X								
	a5				X	X		X				

5. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of dataflow testing, derive different test cases, execute these test cases and discuss the test results.

### ALGORITHM

STEP 1: Define lockPrice=45.0, stockPrice=30.0, barrelPrice=25.0

STEP2: Input locks

STEP3: while(locks!=-1) ,,input device uses -1 to indicate end of data goto STEP 12

STEP4:input (stocks, barrels)

STEP5: compute lockSales, stockSales, barrelSales and sales

STEP6: output("Total sales:" sales)

STEP7: if (sales > 1800.0) goto STEP 8 else goto STEP 9

STEP8: commission=0.10\*1000.0; commission=commission+0.15 \* 800.0;  
commission = commission + 0.20 \* (sales-1800.0)

STEP9: if (sales > 1000.0) goto STEP 10 else goto STEP 11

STEP10: commission=0.10\* 1000.0; commission=commission + 0.15 \*  
(sales-1000.0)

STEP11: Output("Commission is \$", commission)

STEP12: exit

## PROGRAM CODE:

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int locks, stocks, barrels, t_sales, flag = 0;
    float commission;
    clrscr();
    printf("Enter the total number of locks");
    scanf("%d",&locks);
    if ((locks <= 0) || (locks > 70))
    {
        flag = 1;
    }
    printf("Enter the total number of stocks");
    scanf("%d",&stocks);
    if ((stocks <= 0) || (stocks > 80))
    {
        flag = 1;
    }
    printf("Enter the total number of barrelss");
    scanf("%d",&barrels);
    if ((barrels <= 0) || (barrels > 90))
    {
        flag = 1;
    }
    if (flag == 1)
    {
        printf("invalid input");
        getch();
        exit(0);
    }

    t_sales = (locks * 45) + (stocks * 30) + (barrels * 25);

    if (t_sales <= 1000)
    {
        commission = 0.10 * t_sales;
    }
}
```

```
    else if (t_sales < 1800)
    {
        commission = 0.10 * 1000;
        commission = commission + (0.15 * (t_sales - 1000));
    }
    else
    {
        commission = 0.10 * 1000;
        commission = commission + (0.15 * 800);
        commission = commission + (0.20 * (t_sales - 1800));
    }
    printf("The total sales is %d \n The commission is %f",t_sales,
    commission);
    getch();
    return;
}
```

**Test Report:****Precondition :** Enter -1 for locks to exit from input loop**Brief Description :** Enter the locks, stocks and barrels > 0**Define /Use nodes for variables in the commission problem**

Variable name	Define d at node	Used at Node
lprice	7	24
sprice	8	25
bprice	9	26
tlocks	10,16	16,21,24
tstocks	11,17	17,22,25
tbarrels	12,18	18,23,26
locks	13,19	14,16
stocks	15	17
barrels	15	18
lsales	24	27
ssales	25	27
bsales	26	27
sales	27	28,29,33,34,37, 39
comm	31,32, 33,36, 37,39	32,33,37,42



6. Design, develop, code and run the program in any suitable language to implement the binary search algorithm. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results.

### ALGORITHM

Step 1: Input value of „n“. Enter „n“ integer numbers in array int

mid; Step 2: Initialize low = 0, high = n -1

Step 3: until ( low <= high ) do  
    mid = (low + high) / 2  
    if ( a[mid] == key )  
        then do Step 5  
    else if ( a[mid] > key )  
        then do  
            high = mid - 1  
    else  
        low = mid + 1

Step 4: Print unsuccessful search do step 6.

Step 5: Print Successful search. Element found at position mid+1.

Step 6: Stop.

**PROGRAM CODE:**

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int a[20],n,low,high,mid,key,i,flag=0;
    clrscr();
    printf("Enter the value of n:\n");
    scanf("%d",&n);
    if(n>0)
    {
        printf("Enter %d elements in ASCENDING order\n",n);
        for(i=0;i<n;i++)
        {
            scanf("%d",&a[i]);
        }
        printf("Enter the key element to be searched\n");
        scanf("%d",&key);
        low=0;
        high=n-1;
        while(low<=high)
        {
            mid=(low+high)/2;
            if(a[mid]==key)
            {
                flag=1;
                break;
            }
            else if(a[mid]<key)
            {
                low=mid+1;
            }
            else
            {
                high=mid-1;
            }
        }
    }
}
```

```
if(flag==1)
    printf("Successful search\n Element found at Location
    %d\n",mid+1);
else
    printf("Key Element not found\n");
}

else
    printf("Wrong input");
```

```
    getch();  
    return 0;  
}
```





### Viva Questions

1. **What is the MAIN benefit of designing tests early in the life cycle?**  
It helps prevent defects from being introduced into the code.
2. **What is the purpose of exit criteria?**  
The purpose of exit Criteria is to define when a test level is completed.
3. **What determines the level of risk?**  
The likelihood of an adverse event and the impact of the event determine the level of risk.
4. **When is used Decision table testing?**  
Decision table testing is used for testing systems for which the specification takes the form of rules or cause-effect combinations. In a decision table the inputs are listed in a column, with the outputs in the same column but below the inputs. The remainder of the table explores combinations of inputs to define the outputs produced.
5. **What is beta testing?**  
Testing performed by potential customers at their own locations.
6. **We use the output of the requirement analysis, the requirement specification as the input for writing ...**  
User Acceptance Test Cases
7. **What are the benefits of Independent Testing?**  
Independent testers are unbiased and identify different defects at the same time.
8. **In a REACTIVE approach to testing when would you expect the bulk of the test design work to be begun?**  
The bulk of the test design work begun after the software or system has been produced.
9. **What are the phases of a formal review**  
In contrast to informal reviews, formal reviews follow a formal process. A typical formal review process consists of six main steps: Planning, Kick-off, Preparation, Review meeting, Rework, Follow-up.

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

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

**11. What is negative and positive testing?**

A negative test is when you put in an invalid input and receives errors. While a positive testing, is when you put in a valid input and expect some action to be completed in accordance with the specification.

**12. What is the purpose of a test completion criterion?**

The purpose of test completion criterion is to determine when to stop tes

**13. What is the difference between re-testing and regression testing?**

Re-testing ensures the original fault has been removed; regression testing looks for unexpected side effects.

**14. Could reviews or inspections be considered part of testing?**

Yes, because both help detect faults and improve quality.

**15. How much testing is enough?"**

The answer depends on the risk for your industry, contract and special requirements.

**16. Which of the following is the main purpose of the integration strategy for integration testing in the small?**

The main purpose of the integration strategy is to specify which modules to combine when and how many at once.

**17. What is test coverage?**

Test coverage measures in some specific way the amount of testing performed by a set of tests (derived in some other way, e.g. using specification-based techniques). Wherever we can count things and can tell whether or not each of those things has been tested by some test, then we can measure coverage.

**18. What is called the process starting with the terminal modules?**

Bottom-up integration



