Lesson – 4 – Recursion

Recursive algorithm

1. Base Case – Direct solution/stopping condition for the recursion
2. Recursive case – Call itself

Example - Count down – Recursive algorithm

Input n = 5 output : 5 4 3 2 1

```
public static void countdown(int n){

// Base Case

 If(n<=0) // 0<=0

    System.out.println(“Stopped”);

Else

{

     System.out.println(n);   // 5   4    3  2   1

Countdown(n-1); // Recursive case  0

}

}

// Iterative approach

For(int I = 1; i<=5;i++}

   System.out.println(i);
```

Problem : You need to find the length of a String, recursive time

String x = “java”;

1. Method signature
2. Base case
3. Recursive case

Input x == null → base case reach x==null case return 0

Input is “Java” , return 4, recursive will reach the base case of x==””

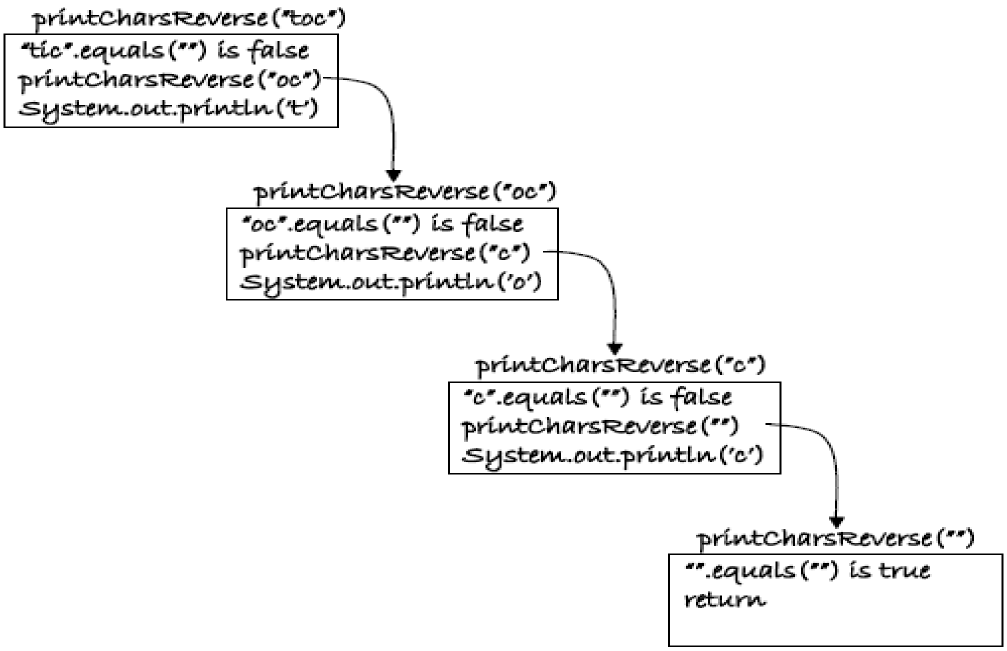
```
Public static int length(String x){
  If(x==null) || if (x=="")
        Return 0;
Else
       Return 1 + length(x.subString(1));
```

Problem: Print the input string in reversed order

Input is “Java” print on console avaj

```
Public static void reverse(String x){
If(x==null) || if (x=="")
        Return; // does not return anything, but return to the call
Else       {
                Reverse(x.subString(x.lenght-1));
               System.out.println(x.charAt(0));
       }
```

Trace the execution of printCharsReverse("toc") using activation frames.



printCharsReverse("toc")
"tic".equals("") is false
printCharsReverse("oc")
System.out.println('t')

printCharsReverse("oc")
"oc".equals("") is false
printCharsReverse("c")
System.out.println('o')

printCharsReverse("c")
"c".equals("") is false
printCharsReverse("")
System.out.println('c')

printCharsReverse("")
"".equals("") is true
return

```
Char lc = x.charAt(x.length-1);
                 System.out.print(lc);
                Reverse(x.subString(0,x.length-2));
}
}
```

Factorial

5! = 5 * 4 * 3 * 2 * 1

Fact(4) = 4 → 4 * 3 * 2 * 1

Fact (5) → 5 * 4 * 3 * 2 * 1

Fact(n) → n * fct(n-1) // Recursive case

If n= 1 or n ==0 output 1 // base case

If input is 2 = 2 * fact(1);

If input is 3 = 3 * fact(2);

If input is n = n * fact(n-1) // Recursive case

Fibonacci Series

F1 = 0 ; F2 = 1   print f1 or f2 / /base case

F3 = f2 + f1;

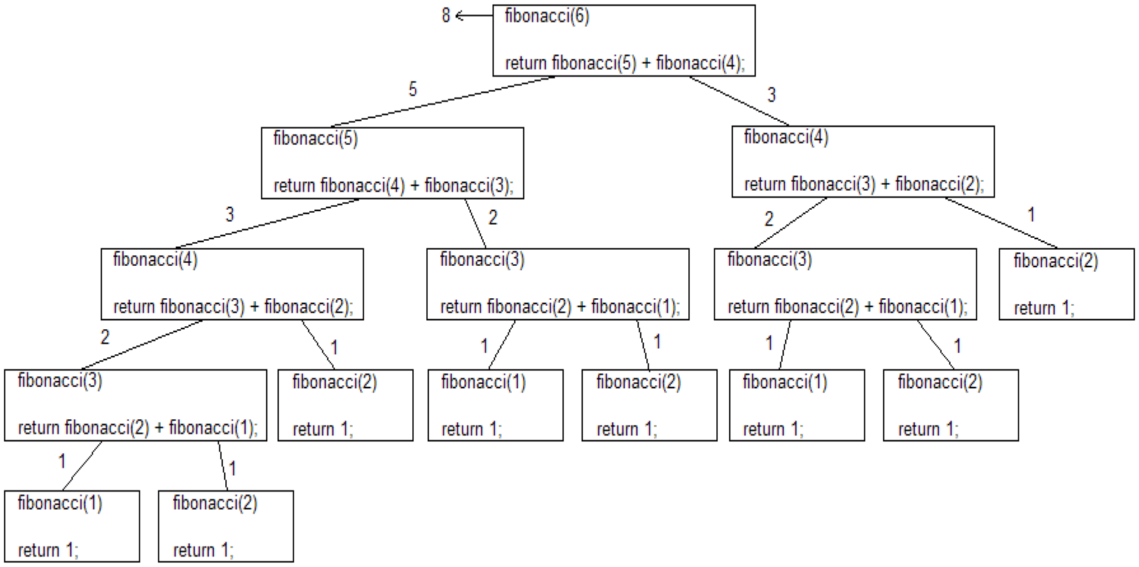
F4 = f3 + f2;

Fn = F(n-1) + f(n-2) // Recursive case

1 1   2   3 5   8 13 ..

Input 3 → 3

Input 6 → 8



Array Searching algorithm

Searching Algorithm – Needs to return the position if the element found or else return -1

1. Linear Search – Search the elements one by one
2. Binary Search –
    a. Your input should be sorted order
    b. Search either in the left side or right side.

Mid = low + high / 2

```
 If tatget < mid  do recursive search on left side
If(target > mid  do recursive search on right side
If target == mid return mid
Else
Return – 1
```

Linear Search

Method signature

```
// Client side call
Public static int LinearSearch(int[] coll, int target) {
    Int rpos = LinearSearch(coll, target,0);
}
// Utility method
Private static int LinearSearch(int[] a, int target, int pos){
If(a[pos] == target
    Return pos;
If(pos == a.length)
   Return -1;
Else
Return    LinearSearch(a,target,pos+1)
}
```

Coll = [12, 15, 23, 55, 5, 6]

Target = 26

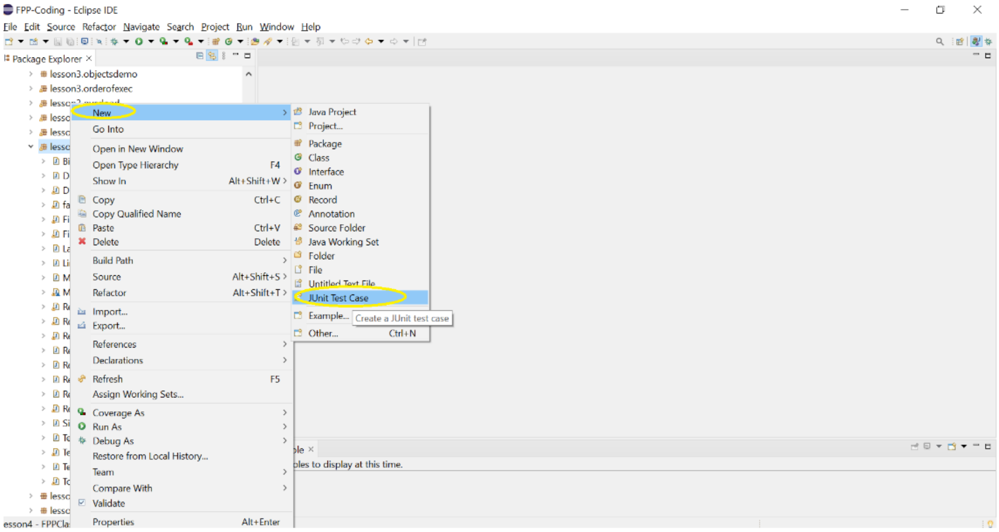
If(target = arr[

Got a match – Return the index or position

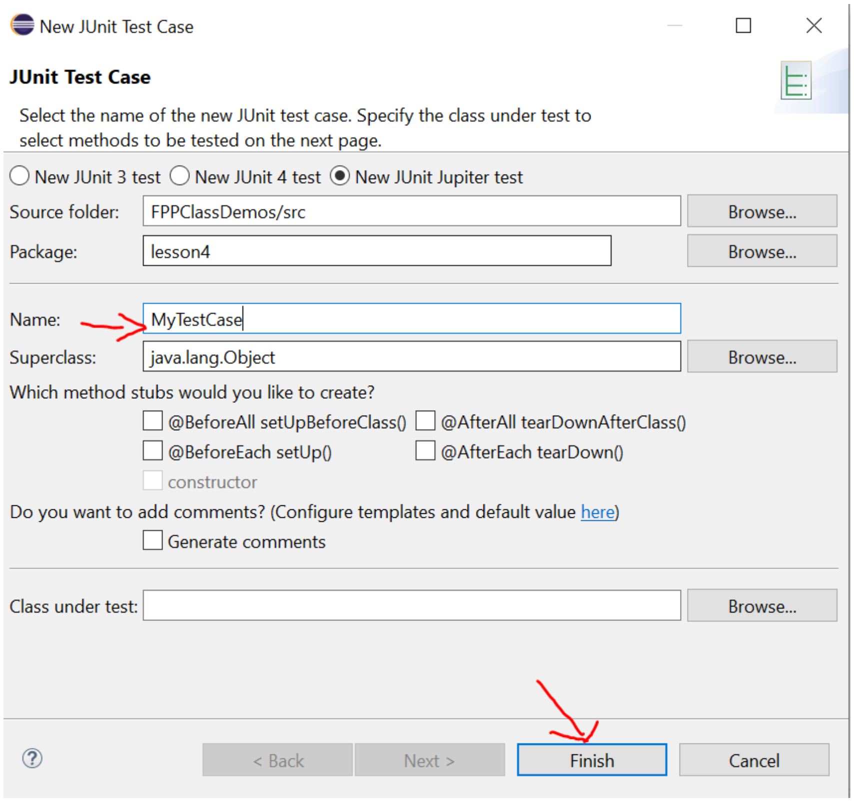
No match → -1

To add Junit Test Case into your Project

Right Click on your Package→ Select



Step 2: Give name for the Test Case and Select Finish


New JUnit Test Case

JUnit Test Case

Select the name of the new JUnit test case. Specify the class under test to select methods to be tested on the next page.

○ New JUnit 3 test ○ New JUnit 4 test ● New JUnit Jupiter test

Source folder: FPPClassDemos/src

Package: lesson4

Name: MyTestCase

Superclass: java.lang.Object

Which method stubs would you like to create?

□ @BeforeAll setUpBeforeClass() □ @AfterAll tearDownAfterClass()

□ @BeforeEach setUp() □ @AfterEach tearDown()

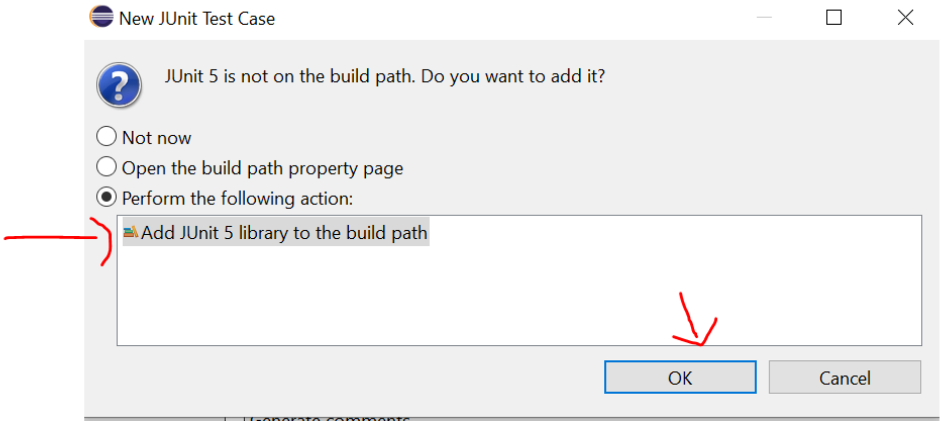
□ constructor

Do you want to add comments? (Configure templates and default value here)

□ Generate comments

Class under test:

Step 3: You will get the below screen to add the Junit library in your project



New JUnit Test Case

JUnit 5 is not on the build path. Do you want to add it?

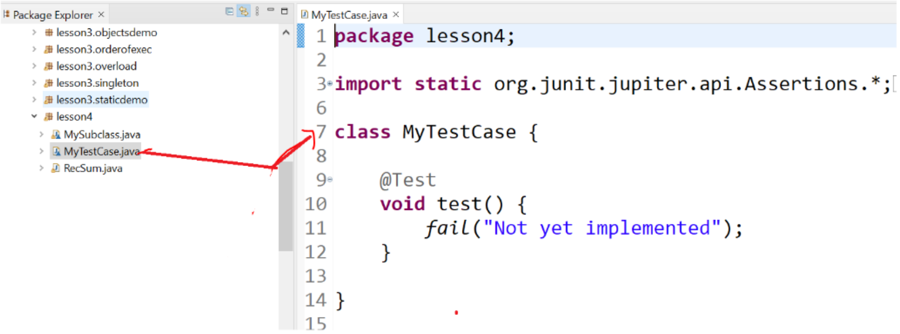
○ Not now

○ Open the build path property page

● Perform the following action:

Add JUnit 5 library to the build path

Step 4: You can see the Junit Test Case file as mentioned below



```
package lesson4;

import static org.junit.jupiter.api.Assertions.*;

class MyTestCase {

    @Test
    void test() {
        fail("Not yet implemented");
    }

}
```