



JOBSHEET 11

Recursive Function

1. Learning Outcomes

- Student must be able to master recursive function concept
- Student must be proficient to implement recursive function as a problem solving option

2. Labs Activity

2.1 Experiment 1

Times: 60 minutes

In this experiment, we will implement a program to calculate factorial number by using **recursive** function. An **iterative**-based factorial program will be developed also, for comparative reason.

1. Create a new Java class and save it as **RecursiveStudentIDNumber.java**
2. Create a new **static** function **factorialRecursive()** that has **int** datatype and 1 **int** parameter. This parameter will be used to pass a value to calculate factorial.

```
static int factorialRecursive(int n){  
    if(n==1)  
        return 1;  
    else  
        return n*factorialRecursive(n-1);  
}
```

3. Create a new **static** function to calculate factorial using iterative approach. Name the function as **factorialIterative()**, with **int** datatype and also has 1 **int** parameter.

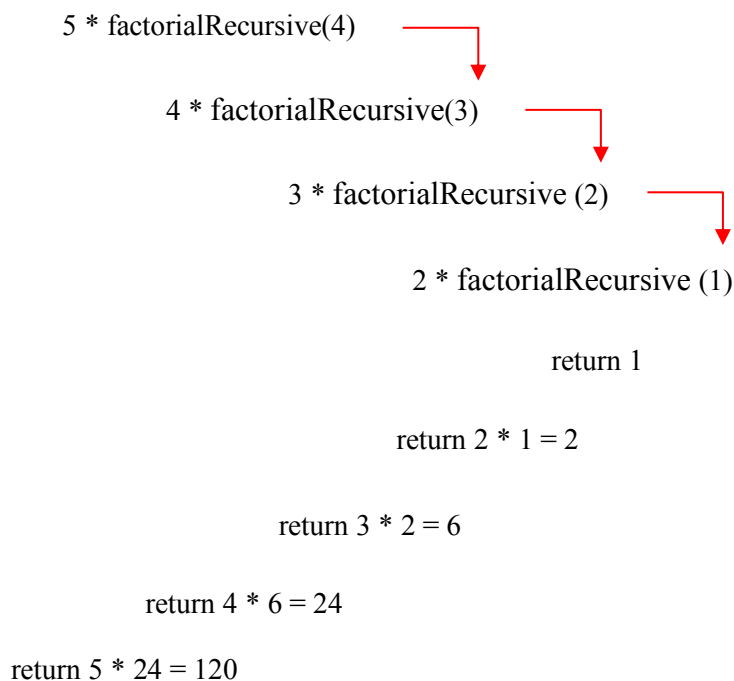
```
static int factorialIterative(int n){  
    int factorial = 1;  
    for(int i=n;i>=1;i--){  
        factorial = factorial*i;  
    }  
    return factorial;  
}
```

4. Create a **main** function to call both of recursive function by passing a value to be calculate the factorial number. Also display the result in this main function.



```
public static void main(String[] args) {
    System.out.println(x:"Hello world!!!");
    System.out.println(factorialRecursive(n:5));
    System.out.println(factorialIterative(n:5));
}
```

5. Run the code and observe the result!
6. When we call function **factorialRecursive(5)**, the following step illustrates how the recursive function called as well as how the result computed (in reverse of recursive calls):



Question

1. After observing the experiment of recursive function above, what is the definition of recursive function?
2. How the recursive function works?
3. From the experiment above, do factorialRecursive() and factorialIterative() have the similar result? Then, what are the differences between recursive and iterative if both are having the same result?

2.2 Experiment 2

Times : 60 minutes

In the following experiment, a program to calculate power of a number using recursive function.



1. Create a new Java class, and name it as **PowerRecursiveIDStudentNumber.java**
2. Create a new **static** function **calculatePower()** that has **int** datatype with 2 int parameters (base and power number).

```
static int calculatePower(int base, int pow){  
    if(pow==0)  
        return 1;  
    else  
        return base*calculatePower(base, pow-1);  
}
```

3. Create a new **main()** function that will get user input for base number and power number, will call the recursive function to calculate the power result and will display the final result.

```
Scanner input = new Scanner(System.in);  
System.out.print(s:"Input Base Number: ");  
int base = input.nextInt();  
System.out.print(s:"Input Power Number: ");  
int power = input.nextInt();
```

4. Call function **calculatePower()** and do not forget to pass 2 values into parameter, otherwise the error will arises.

```
System.out.println("Result of "+base+" power "+power+" = "+  
    calculatePower(base, power));
```

5. Run the program and observer the result!

Question!

1. In Experiment 2, there is a recursive function call **calculatePower()** in the main function, then the function **calculatePower()** is called repeatedly. Explain how long the function calling process will run!
2. Add program code to print the power calculation series. Example:
calculatePower(2,5) will print 2x2x2x2x2x1 = 32!

2.3 Experiment 3

Time : 60 minutes

We will continue this experiment by creating a new program to calculate the amount of investor money that will be used to invest after earning profits for several years. This



program must implement recursive function to calculate the profits that would be got by investor.

1. Create a new class with name **ProvitRecursiveStudentIDNumber.java**
2. Create a new **static** function **calculateProfit()** that has **double** datatype and 2 **int** parameters. The **balance** would be passed through first parameter and **investment period** would be passed through parameter 2. The investor will get 11% interest annually.

Since to calculate profit is **interest * balance**, so the total amount of investor money will be **balance+interest*balance**. In this case, the amount of **interest** is **0.11 * balance**, and the final balance is considered as **1*balance**, so that **1 * balance + 0.11 *** or equivalent formula to calculate the final balance will be **1.11 * balance** in a year.

```
static double calculateProfit(double balance, int period){  
    if(period==0)  
        return balance;  
    else  
        return 1.11*calculateProfit(balance, period-1);  
}
```

3. Create a new **main()** function, then get the user input for initial **balance** and **investment period**.

```
System.out.print(s:"Input Balance: ");  
double initialBalance = input.nextInt();  
System.out.print(s:"Input Investment Period: ");  
int investPeriod = input.nextInt();
```

4. Then call function **calculateProfit()** from main function.

```
System.out.println("Balance after "+base+" year = "+  
    calculateProfit(initialBalance, investPeriod));
```

5. Run the program and observe the result!

Question!

1. From the above experiment, which statements that is classified as “**base case**” and “**recursion call**”!
2. Explain using simulation or trace the expansion phase and substitution phase of **calculateProfit(100000,3)** function!

3. Assignment

Times: 120 minutes

1. Write a program to display numbers n to 0 using recursive functions and iterative functions. (**DescendingSequenceRecursive**).
2. Create a program to sum the numbers using recursive function. For example $n = 8$, then it will result $1+2+3+4+5+6+7+8 = 36$ (**SummationRecursive**).
3. Create a program that contains a recursive function to check whether a number n is a prime number or not. A number will not be classified as a prime number if it is divisible by a number less than n . (**PrimeCheckingRecursive**).
4. A pair of newborn guinea pigs (male and female) are placed in a nursery. After two months the guinea pig pair gave birth to a pair of twin guinea pigs (male and female). Every pair of guinea pigs that is born will also give birth to a pair of guinea pigs every 2 months. How many pairs of guinea pigs were there at the end of the 12th month? Write a program using a recursive function! (**Fibonacci**).

And the following table illustrates the calculation.

Month	Pair Number		Pair Total
	Productive	Non-Productive	
1	0	1	1
2	0	1	1
3	1	1	2
4	1	2	3
5	2	3	5
6	3	5	8
7	5	8	13
8	8	13	21
9	13	21	34
10	21	34	55
11	34	55	89
12	55	89	144