Functions 1

Content of this lecture

1. Factorial
2. Function Intro
3. Sum()
4. Quizzes
5. Power function
6. Ceil/floor
7. Assignments

Factorial

Given an non negative integer N, the factorial of N is defined as follows:

Factorial of N = Product of all numbers from `1` to `N`.

Factorial	Output
4!	4 * 3 * 2 * 1 = 24
3!	3 * 2 * 1 = 6

Question

What is the factorial of 5?

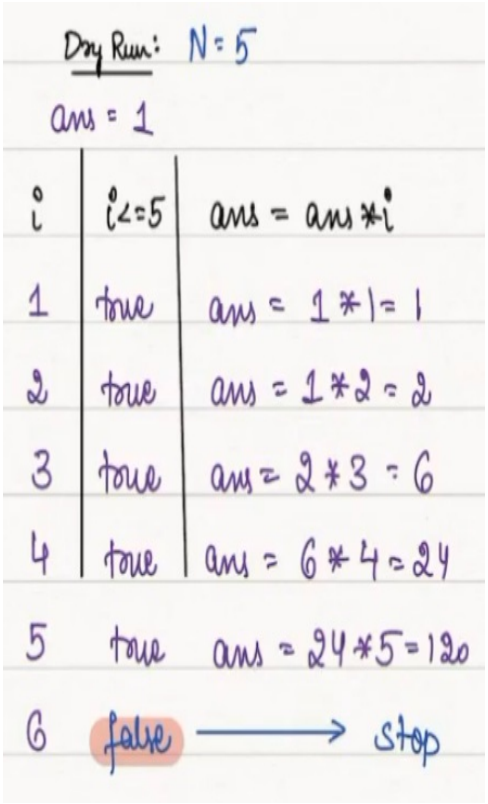
Choices

- [ ] 5
- [ ] 15
- [ ] 24
- [x] 120

Code:

```
public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int N = scanner.nextInt();
        // multiply numbers from 1 to N
        int ans = 1;
        for(int i = 1;i <= N;i ++ ){
        ans *= i;
        }
        S.O.Pln(ans);
    }
```

Dry run for N = 5:



---

nCr

$^{N}C_{R}$: Given N and R (where N>=R) calculate $^{N}C_{R}$

$$^{N}C_{R} = \frac{N!}{R!\,(N-R)!}$$

N	R	Output
5	2	$^{5}C_{2} \rightarrow \frac{5!}{2! * 3!} = \frac{1*2*3*4*5}{1*2 * 1*2*3} = 10$
4	3	$^{4}C_{3} \rightarrow \frac{4!}{3! * 1!} = \frac{1*2*3*4}{1*2*3*1} = 4$

Dry Run for some values of N and R:

Question

$^{6}C_{2}$

Calculate

Choices

- ☐ 10

- [x] 15
- [ ] 20
- [ ] 25

Explanation:

$$6 \mid 2 \mid {}^{6}C_{2} \mid \frac{6!}{2! \times 4!} = \frac{1 \times 2 \times 3 \times 4 \times 5 \times 6}{1 \times 2 \times 1 \times 2 \times 3 \times 4} = 15$$

Small Question:

> How many inputs are required to calculate NCR? -- Ans = 2

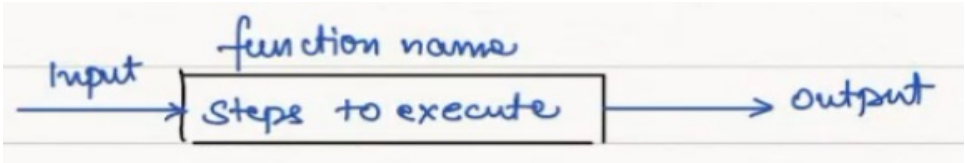
Code for calculating NCR:

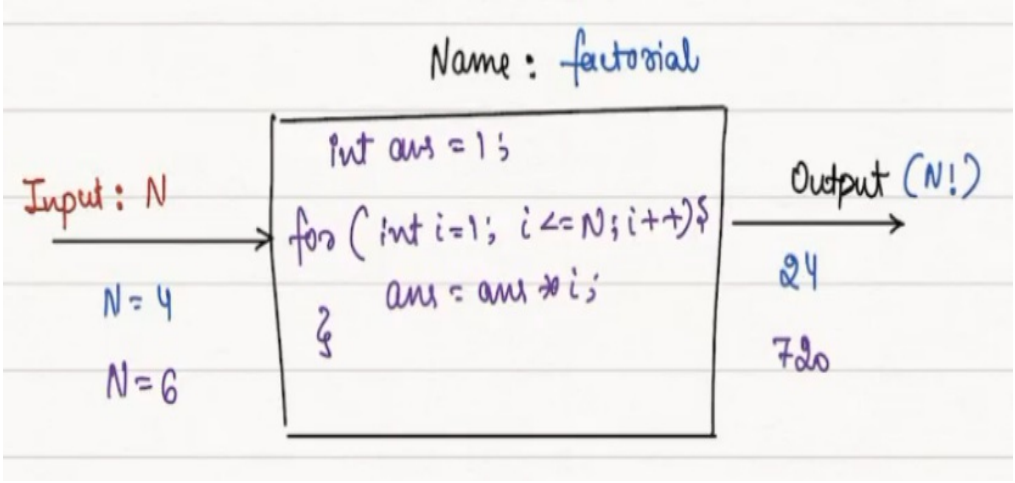
```
public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int N = scanner.nextInt();
        int R = scanner.nextInt();
        int nf = 1;
    // calculate N!
        for(int i = 1;i <= N;i ++ ){
            nf *= i;
        }
    // calculate R!
        int rf = 1;
        for(int i = 1;i <= R;i ++ ){
            rf *= i;
        }
    // calculate (N-R)!
        int nrf = 1;
        for(int i = 1;i <= (N - R);i ++ ){
            nrf *= i;
        }
        S.O.Pln(nf / (rf * nrf));
    }
```

Functions

> Give a real-life analogy. Suppose you need a screw-driver. First time you will buy that screw driver. From next time onwards you will use the same screw-driver.

Structure of a function:





Diagram/Flow Chart for Factorial function:

What is the need of a function? To reuse the code, i.e., we dont need to write the same code again and again.

Syntax of a function:

```
output_type function_name(data_type input_name){
    // function body
    // i.e. the code to be executed
    return ans; // the return keyword is used to return the output from the function
}
```

Writing the first function:

1. You cannot write a function inside the `main()`.
2. Write it before `public static void main()` and inside the `class Main{}`.
3. In Java, methods are also known as functions.

Factorial function:

```
static int factorial(int a){
    int ans = 1;
    for(int i = 1; i <= a; i ++ ){
        ans = ans * i;
    }
    return ans;
}
```

Calling the factorial function from the main function:

```
public static void main(){
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    System.out.println(factorial(5));
}
```

Output:

```
120
```

We cannot call non static functions from inside static functions. Since the main function is static, so to call the factorial function from inside the main function, we need to make the factorial function as static by writing the keyword `static` in front of it.

Write a function that returns sum of two numbers:

```
static int sum2(int a, int b){
    int sum = a + b;
    return sum;
}
```

> Question: What will be the output of `sum2(10,5)` -- Ans = 15

Code

```
1 public class MainClass {
2   public static int sum2(int a, int b){
3     int sum = a + b;
4     return sum;
5 }
6   public static void main(String[] args) {
7   System.out.print(sum2(10,20));
8   }
9 }
```

Write a function that returns sum of three numbers:

```
static int sum3(int a, int b, int c){
    return a + b + c;
}
```

Question

What is the output of the following code?

```
static int sum(int a, int b){
    return a + b;
}

public static void main(String args[]){
    sum(5,10);
}
```

Choices

- [ ] 15
- [ ] Error
- [x] None

Explanation: This is because we are not printing the answer in the code.

Question

What is the output of the following code?

```
static int sum(int a, int b){
    return a + b;
}

public static void main(String args[]){
    System.out.println(sum(5,10));
}
```

Choices

- [ ] Error
- [x] 15
- [ ] No output

Write a function that returns the max of two numbers:

```
static void max_of_two(int a, int b){
    int max = 0;
    if(a > b){
        max = a;
    }
    else{
        max = b;
    }
    SOPln(max);
}
```

Calling this function:

```
System.out.println(max_of_two(5,9));
```

Output:

```
9
```

Code

```
1 public class MainClass {
2   public static void max_of_two(int a, int b){
3     int max = 0;
4     if(a > b){
5         max = a;
6     }
7     else{
8         max = b;
9     }
10     System.out.println(max);
11 }
12   public static void main(String[] args) {
13   max_of_two(10,20);
14   }
15 }
```

Write a function that will print the product of two numbers:

> Stress that the function needs to print the value rather than returning it.

```
static void product_of_two(int a, int b){
    int product = a * b;
    System.out.println(product);
}
```

> The return type of the function changes from `int` to `void`. `void` data type is used to denote that the function is not returning anything.

Calling this function:

```
product_of_two(5,9);
```

Output:

```
45
```

Question

What is the output of the following code?

```
static void sum(int a, int b){
    return a + b;
}

public static void main(String args[]){
    System.out.println(sum(5,10));
}
```

Choices

- [x] Error
- [ ] 15
- [ ] No output

Explanation: The return type of the function is wrong. You are returning an int, but the return type is set to void.

Question

What is the output of the following code?

```
static int extra(int a, int b){
    return a + b + 10;
}

public static void main(String args[]){
    int ans = extra(5,10);
    System.out.println(ans);
}
```

Choices

- [ ] 15
- [x] 25
- [ ] Error
- [ ] No output

Question

What is the output of the following code?

```
static int square(int a){
    return a * a;
}

public static void main(String args[]){
    System.out.println(square( - 6 ));
}
```

Choices

- [ ] -36
- [x] 36
- [ ] 6
- [ ] Error

Question

What is the output of the following code?

```
public static int square(int x){
    return x * x;
}

public static int add(int x, int y){
    return x + y;
}

public static void main(String args[]){
    int n1 = square(3);
    System.out.println(add(n1,square(9)));
}
```

Choices

- [ ] 81
- [ ] 9
- [x] 90
- [ ] 115

Explanation: n1 = square(3) = 9 square(9) = 81 add(9,81) = 90

Ceil Function

Given a number x, return smallest number >= x

- ceil(3.42) = 4
- ceil(5.67) = 6

Question

What is the value of `ceil(6.5)`?

Choices

- [ ] 6
- [x] 7
- [ ] 8
- [ ] 9

Question

What is the value of `ceil(6)`?

Choices

- [ ] 7
- [x] 6
- [ ] 5
- [ ] 4

Question

What is the value of `ceil(-3.4)`?

Choices

- [x] -3
- [ ] -2
- [ ] -4
- [ ] -1

Question

What is the value of `ceil(-4.5)`?

Choices

- [ ] -3
- [x] -4
- [ ] -5
- [ ] -6

Floor Function

Given a number x, return the greatest integer that is $<= x$.

- floor(3.2) = 3
- floor(5.67) = 5
- floor(6.5) = 6
- floor(-3.4) = -4
- floor(-4.5) = -5
- floor(-7) = -7

Predefined functions

Functions which are already defined in the coding languages to make life simpler. Rules for using predefined functions.

1. Check parameters
2. Check return type

Examples:

1. `Math.ceil()`
2. `Math.floor()`
3. `Math.power(a,b)`

Predefined function	Number of inputs	Return type
Math.ceil	1	double
Math.floor	1	double
Math.power	2	double

Some more pre-defined functions:

> Explain there workings with small examples

- `Math.random()`
- `Math.abs()`
