**Assignment 6**

**- 이름:**

1. 다음 실행 결과를 출력하는 프로그램을 **for문**을 사용해서 작성하시오.

|  |
| --- |
| **\***  **\*\***  **\*\*\***  \*\*\*\*  \*\*\*\*\* |

*public* *class* Assignment6 {

*public* *static* void *main*(String[] args) *throws* Exception {

*for* (int i = 1; i < 6; i++) {

*for* (int j =0; j<i; j++){

                System.*out*.*print*("\*");

            }

            System.*out*.*println*("");

        }

    }

}

2. **메소드 오버로딩 개념**과 **while문**을 활용하여 팩토리얼을 구현하는 코드를 작성하시오.

메소드의 이름은 factorial이고, 아래와 같이 사용 가능하다.

**factorial(6) = 720** 값을 출력 (1\*2\*3\*4\*5\*6 = 720 이기 때문에)

**factorial(3, 6) = 360** 값을 출력 (3\*4\*5\*6 = 360 이기 때문에)

즉, 위와 같은 기능을 하도록 factorial 메소드를 구현하고,

**factorial(10)**과 **factorial(4, 10)**에 대한 값을 제대로 계산하는지 출력하시오.

(코드와 출력 결과 등 첨부)

텍스트이(가) 표시된 사진

자동 생성된 설명

*public* *class* Assgnment66 {

*public* *static* void *main* (String[] args) {

    int n = 10, n1 = 4, n2 = 10;

    System.*out*.*printf*("factorial(%d) = %d%n", n, *factorial*(n));

    System.*out*.*printf*("factorial(%d, %d) = %d", n1, n2, *factorial*(n1, n2));

  }

*public* *static* int *factorial*(int n) {

*if* (n <= 1)

*return* n;

*else*

*return* *factorial*(n-1) \* n;

  }

*public* *static* int *factorial*(int n1, int n2) {

*if* (n1 <= 1)

*return* n1;

*else*

*return* *factorial*(n2-1) \* n1;

  }

}