Fundamentals of Object Oriented Programming

CSN- 103

Dr. R. Balasubramanian

Associate Professor

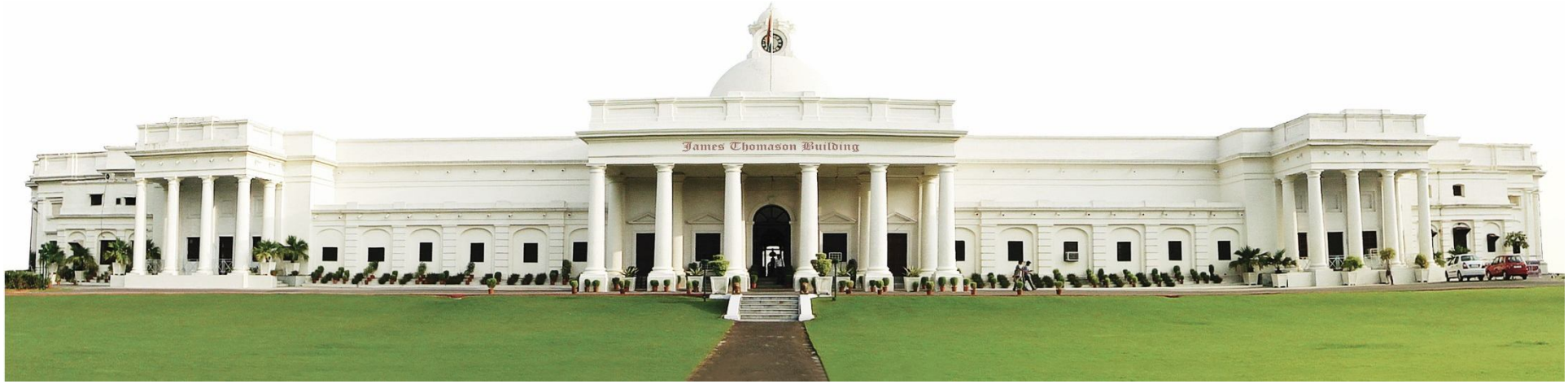
Department of Computer Science and Engineering

Indian Institute of Technology Roorkee

Roorkee 247 667

balarfcs@iitr.ac.in

https://sites.google.com/site/balaiitr/



Recursion in JAVA

- Write a program to find a factorial of a given number using recursive method.

$$n! = (n-1)! \times n$$

$n! = (n-1)! \times n$

$3!$

fact(3) $n = 3$ → fact(2) * 3 → 2 * 3 → 6

fact(2) → fact(1) * 2 → 2

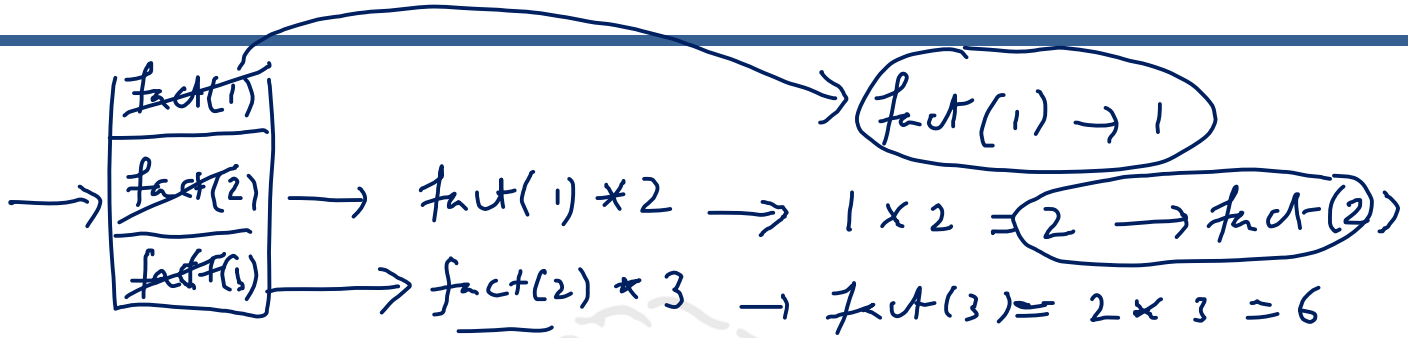
```
1  class Factorial {
2      int fact(int n) {
3          int result;
4          if ( n ==1) return 1;
5          result = fact (n-1) * n;
6          return result;
7          }
8  }
9
10 class Recursion {
11     public static void main (String args[]) {
12         Factorial f =new Factorial();
13         System.out.println("Factorial of 3 is " + f.fact(3));
14         System.out.println("Factorial of 4 is " + f.fact(4));
15         System.out.println("Factorial of 5 is " + f.fact(5));
16     }
17 }
```

default constructor

Terminal

```
sh-4.3$ javac Recursion.java
sh-4.3$ java Recursion
Factorial of 3 is 6
Factorial of 4 is 24
Factorial of 5 is 120
sh-4.3$
```

Stack


fact(1)
fact(2)
fact(1)
fact(1) → 1
fact(1) * 2 → 1 x 2 = 2 → fact(2)
fact(2) * 3 → fact(3) = 2 x 3 = 6

```
1  class Recursion {
2      public static void main (String args[]) {
3          Factorial f =new Factorial();
4          System.out.println("Factorial of 3 is " + f.fact(3));
5          System.out.println("Factorial of 4 is " + f.fact(4));
6          System.out.println("Factorial of 5 is " + f.fact(5));
7      }
8  }
9
10 class Factorial {
11     int fact(int n) {
12         int result;
13     if ( n ==1) return 1;
14     result = fact (n-1) * n;
15     return result;
16     }
17 }
```

```
Terminal
sh-4.3$ javac Recursion.java
sh-4.3$ java Recursion
Factorial of 3 is 6
Factorial of 4 is 24
Factorial of 5 is 120
sh-4.3$
```

Recursion in JAVA

```
1  class Easyfun {
2        void easy(int n) {
3           if ( n < 1) return;
4        easy(n-1);
5        System.out.print(n);
6        easy(n-2);
7        }
8  }
9
10 class Testeasy {
11       public static void main (String args[]) {
12           Easyfun e =new Easyfun();
13            e.easy(5);
14            }
15 }
```

Terminal

```
sh-4.3$ javac Testeasy.java
sh-4.3$ java Testeasy
123141251231sh-4.3$
```

Recursion in JAVA

```
1  class Easyfun {
2        void easy2(int n) {
3           if ( n < 1) return;
4        easy2(n-1);
5        easy2(n-2);
6        System.out.print(n);
7        }
8  }
9
10 class TestEasy2{
11       public static void main (String args[]) {
12            Easyfun e =new Easyfun();
13             e.easy2(5);
14             }
15 }
```

Terminal

```
sh-4.3$ javac TestEasy2.java
sh-4.3$ java TestEasy2
121312412135sh-4.3$
```

Recursion in JAVA

```
1  class Findout{
2  void findoutput(int num)
3  {
4   if (num < 1) return;
5    findoutput(num / 2);
6   System.out.print(num % 2);
7    }
8  }
9
10 class Recursion1 {
11       public static void main (String args[]) {
12            Findout d =new Findout();
13            d.findoutput(20);
14       }
15 }
```

Terminal

```
sh-4.3$ javac Recursion1.java
sh-4.3$ java Recursion1
10100sh-4.3$
```

Find fun(4,3)

```
int fun(int a, int b)
{
    if (b == 0)
        return 1;
    if (b % 2 == 0)
        return fun(a*a, b/2);

    return fun(a*a, b/2)*a;
}
```