



Experiment-2

Name: Deepanshu Saini
Section : 23BCS_KRG_01a
Subject Code:23CSH-301

UID: 23BCS13189
Subject Name : DAA
Date:30/07/2025

1. **Aim:** To implement an efficient algorithm for computing x^y (x raised to the power y) using Exponentiation by Squaring.

2. Objective:

- To understand and apply the concept of **exponentiation by squaring**.
- To reduce the time complexity of computing powers compared to the naïve $O(y)$ approach.
- To write clean, efficient code in Java for power calculation.

3. PseudoCode:

Algorithm (Exponentiation by Squaring):

Initialize output = 1.

Repeat while $y > 0$:

- If y is even:
 - Set $x = x * x$.
 - Set $y = y / 2$.
- Else (y is odd):
 - Set output = output * x.
 - Decrement y by 1.

Return output

4. Code:

```
import java.util.*;

class exp2 {

    public static int power(int x, int y) {

        int output = 1;
```

```
    if (y == 0) {  
        return 1;  
    }  
    while (y > 0) {  
        if (y % 2 == 0) {  
            x = x * x;  
            y = y / 2;  
        } else {  
            output = output * x;  
            y--;  
        }  
    }  
    return output;  
}  
  
public static void main(String[] args) {  
    System.out.println(power(3, 5));  
}  
}
```

5. Output:



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

6. Time Complexity: $O(\log y)$

7. Learning Outcomes:

- How to implement Exponentiation by Squaring for efficient power computation.
 - Difference between naïve and optimized power algorithms.
 - How logarithmic algorithms improve performance for large inputs.
 - Writing clean, concise Java code for mathematical operations.
 - Importance of control flow (if-else, loops) in optimizing computation.
-