```
public class DiffieHellman {

    // Method to calculate modular exponentiation
    private static int calculatePower(int base, int exponent, int modulus) {
        return (int) Math.pow(base, exponent) % modulus;
    }

    public static void main(String[] args) {
        // Public parameters
        int prime = 23; // P
        int generator = 5; // G

        // Private keys
        int privateKeyAlice = 6; // a
        int privateKeyBob = 15; // b

        // Public keys
        int publicKeyAlice = calculatePower(generator, privateKeyAlice, prime); // G^a % P
        int publicKeyBob = calculatePower(generator, privateKeyBob, prime); // G^b % P

        // Shared secret keys
        int secretKeyAlice = calculatePower(publicKeyBob, privateKeyAlice, prime); // (G^b)^a % P
        int secretKeyBob = calculatePower(publicKeyAlice, privateKeyBob, prime); // (G^a)^b % P

        // Output
        System.out.println("Prime (P): " + prime);
        System.out.println("Generator (G): " + generator);
        System.out.println("Alice's private key (a): " + privateKeyAlice);
        System.out.println("Bob's private key (b): " + privateKeyBob);
        System.out.println("Secret key for Alice: " + secretKeyAlice);
        System.out.println("Secret key for Bob: " + secretKeyBob);
    }
}
```