4. Write a program to implement Push & Pop operations using class.

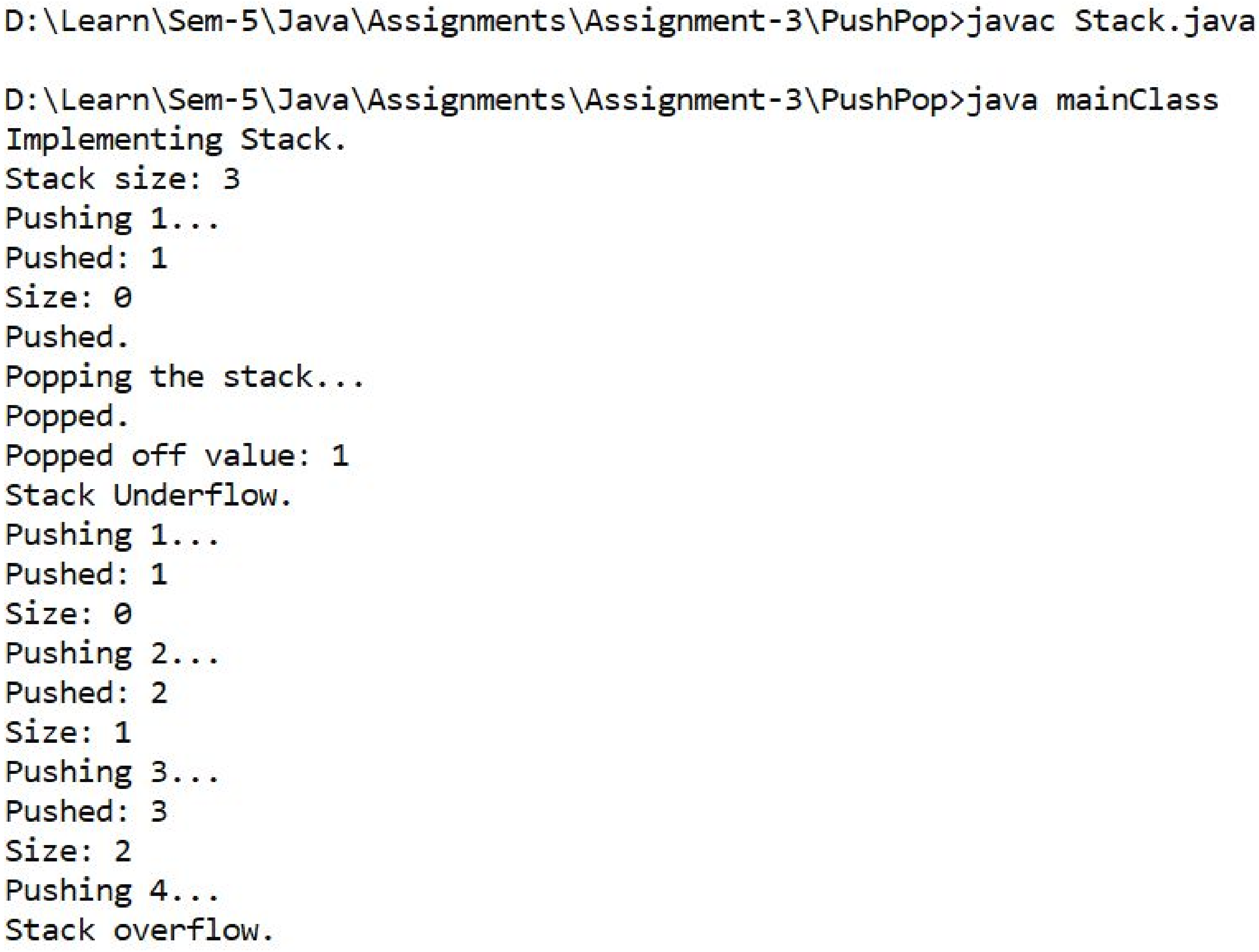
```
class Stack{
      int size;
      static int top = -1;
      int[] array;
      Stack(int size){
            this.size = size;
            array = new int[size];
      }
      void push(int element){
            if(top==size - 1)
                  System.out.println("Stack overflow.");
            else
            {
                  array[++top] = element;
                  System.out.println("Pushed: " + element);
                  System.out.println("Size: " + top);
            }
      }
      int pop(){
            if(top == -1)
            {
                  System.out.println("Stack Underflow.");
                  return 0;
            }
            else
            {
                  System.out.println("Popped.");
                  return array[top--];
            }
      }
}

class mainClass{
      public static void main(String args[]){
            System.out.println("Implementing Stack.");
            System.out.println("Stack size: 3");
            Stack s = new Stack(3);
            System.out.println("Pushing 1...");
            s.push(1);
            System.out.println("Pushed.");
            System.out.println("Popping the stack...");
            int a = s.pop();
```

```
        System.out.println("Popped off value: " + a);
        a = s.pop();
        System.out.println("Pushing 1...");
        s.push(1);
        System.out.println("Pushing 2...");
        s.push(2);
        System.out.println("Pushing 3...");
        s.push(3);
        System.out.println("Pushing 4...");
        s.push(4);

    }
}
```

Output:



```
D:\Learn\Sem-5\Java\Assignments\Assignment-3\PushPop>javac Stack.java

D:\Learn\Sem-5\Java\Assignments\Assignment-3\PushPop>java mainClass
Implementing Stack.
Stack size: 3
Pushing 1...
Pushed: 1
Size: 0
Pushed.
Popping the stack...
Popped.
Popped off value: 1
Stack Underflow.
Pushing 1...
Pushed: 1
Size: 0
Pushing 2...
Pushed: 2
Size: 1
Pushing 3...
Pushed: 3
Size: 2
Pushing 4...
Stack overflow.
```