# Object-Oriented Programming Lab#2

## Today's Topics

- Flow Controls: If, While, For, Do-While
- Recursion
- User Input
- Array

## Code to read user input using Scanner:(need to import java.util.Scanner)

```
Scanner scan = new Scanner (System . in ) ;
int inputNum  =  scan.nextInt();
double input = scan.nextDouble();
```

## Code to read user input using JOptionPane: (need to import javax.swing.JOptionPane)

**static String showInputDialog**(Component parentComponent, Object message)

String name = JOptionPane.*showInputDialog*(***null, "enter name");***

## Problems/Assignments

1. Write a java program to determine whether **a given number is even or odd.**

| Sample Input | Expected Output |
|---|---|
| 7 | odd |
| 8 | even |
| 11 | odd |

2. Write a java program to determine whether **a given number is prime or not**.

| Sample Input | Expected Output |
|---|---|
| 7 | Prime |
| 9 | Not Prime |
| 11 | Prime |

3. Write a program that will take **n integer numbers into an array**, and then **sum** up all the **even integers** in that array.

| Sample input | Sample output |
|---|---|
| {1, 2, 3, 4, 5} | 6 |
| {2, 8, 3, 9, 0, 1} | 10 |

4. Write a program in java to find the factorial of n. Use **recursion**.

| **Sample Input** | **Expected Output** |
|---|---|
| 2 | 2 |
| 3 | 6 |
| 0 | 1 |
| -2 (any negative number) | NA |

5. Write a program in java to display the individual digits of a number. Use **recursion**.

| **Sample Input** | **Expected Output (better version)** | **Expected Output** |
|---|---|---|
| 172 | 1,7,2 | 2, 7, 1 |
| 90357 | 9,0,3,5,7 | 7, 5, 3, 0, 9 |
| 110 | 1,1,0 | 0, 1, 1 |

6. Write a Java program to show the number pyramid. The program will prompt the user to enter an integer number as input. If the **user input is 9 or less, display that many lines** of the pyramid. If **user input is 10 or above, display only 9 lines**. Don't hard code the values, use logic.

| **Sample Input** | **Expected Output** |
|---|---|
| 5 | 1<br>1 2 1<br>1 2 3 2 1<br>1 2 3 4 3 2 1<br>1 2 3 4 5 4 3 2 1 |
| 12 | 1<br>1 2 1<br>1 2 3 2 1<br>1 2 3 4 3 2 1<br>1 2 3 4 5 4 3 2 1<br>1 2 3 4 5 6 5 4 3 2 1<br>1 2 3 4 5 6 7 6 5 4 3 2 1<br>1 2 3 4 5 6 7 8 7 6 5 4 3 2 1<br>1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1 |