Iteration part 2 - `for loop`

Objectives

- Recap previous session
- Introduction to `for` loop instruction
- Class exercises
- Homework exercises
- Guidelines

Recap previous session

- What is the increment operator?
- What is the decrement operator?
- What is the difference between a `postfix increment operator` and `prefix increment operator`?
- In how many ways can you write: `a = a + 1`?
- What do we mean by iteration?
- What is the `while` instruction? What does it do?
- In what scenarios do we need iteration?
- What is an `infinite loop`?

Introduction to `for` loop instruction

```
int i = 0;
while (i <=n) {
    // do something
    i++;
}
```

- Because most of the time, when we have a `while` loop, the condition depends on some variable which has been previously defined and then incremented in the `while` loop's body instruction called the `for loop`
- The `for` loop can be seen as a shorthand notation for the `while` loop and most of the time, what you can achieve with a `while` loop, you can also achieve with the `for` loop
- We typically use the for loop to execute a block of statements a given number of times.
- Let's suppose you want to display the numbers from 1 to 10. Instead of writing ten statements that contains a call to the `println` method, we can write it like this:

```
for (int count = 1; count <=10; count++) {
    System.out.println(count);
}
```

- The syntax of a `for` loop is:

```
    for(init_variable; condition; increment_variable){
      //run the action
  }
```

  - Now, let's see describe each part:
    - `init_variable` - this is the part which executes only once, at the begginning of the loop.
      - it is used for initializing the counter, a.k.a the number which determines how many time the loop should run
    - `condition` - this is the part which is evaluated at the begginning of each loop
      - if it evaluates to `true` then the loop continues
      - if it evaluates to `false` then the loop ends
    - `increment_variable` - this is the part which runs at the end of each cycle.
      - By cycle we mean when all the statements inside the `for`'s body have been executed.
      - Here, this part should increment the counter, or the variable which says how many time this loop will run.
  - Now let's suppose a small example which computes the sum of number from 1 to 100:

```
    public class Application {

        public static void main(String[] args) {
            int sum = 0;
            for(int i = 1; i <=100; i++) {
                sum += i;
            }
            System.out.println("Sum is: "  + sum);
        }
    }
```

  - Let's examine each piece of the previous snippet:
    - The expression `int i = 1` is executed only once, when the loop starts
      - if we attempt to reference the `i` variable outside the scope of the `for`, the code will not compile as this variable only lives inside the scope of the `for`
    - The expression `i <= 100` will be checked at each iteration
      - an `iteration` is simply one exection of a loop scope
        - if, for example, our `for` loop runs for 10 times, we say that we had `10 iterations`
      - Note that after the initialize part, we can have any expression as long as it evaluates to a boolean result
    - The expression `i++` executes after each run of the statements inside the scope of the loop.
  - Also very important, keep in mind that each expression in the header of a for loop should be preceded by a semicolon(`;`):

```
for(init;condition;increment)
```

- We are not forced to only use increment in the last part of the for loop header, we can also use a decrement operator.
- For example, let's suppose the following example when we want to display the first 10 integer numbers, greater than 0, in descending order:

```
public class Application {

    public static void main(String[] args) {

        for(int i = 10; i >=0; i--) {
            System.out.println(i + " ");
        }
    }
}
```

  - As you can see, we are decrementing the value of `i` after each iteration.
  - We can use whatever expression we want as long as at each step, it brings us closer to make the condition evaluate to `false`, otherwise, we will get an `infinite loop`.

Class exercises:

1. Write a JAVA program which displays the first `n` natural numbers, greater than 0, in ascending order. `n` is read from the keyboard
   - Sample Input: 4
   - Sample Output: 1 2 3 4
   - Solution:

```
import java.util.Scanner;

public class Application {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("N: ");
        int n = in.nextInt();

        for(int i = 1; i <= n; i++) {
            System.out.print(i + " ");
        }
    }
}
```

2. Create a JAVA program which will read n numbers from the Standard Input and then will display on one line the first n natural numbers not equal to 0 in ascending order and on another line, the same numbers but in descending order.
    - Sample Input: 5
    - Sample Output:
        - 1 2 3 4 5
        - 5 4 3 2 1
        - Solution:

```
import java.util.Scanner;

public class Application {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("N: ");
        int n = in.nextInt();

        for(int i = 1; i <= n; i++) {
            System.out.print(i + " ");
        }

        System.out.println();

        for(int i = n; i > 0; i--) {
            System.out.print(i + " ");
        }
    }
}
```

3. Create a JAVA program wich will compute the average of n numbers (n read from the keyboard).
    - Sample Input: n = 5
        - 2 5 78 9 11
    - Sample Output: 21.0
    - Solution:

```
import java.util.Scanner;

public class Application {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("How many numbers will you read?  ");
        int n = in.nextInt();
        double sum = 0;
        for(int i = 0; i < n; i++) {
            System.out.print("Enter number #" + (i+1) + ": ");
            int number = in.nextInt();
```

```
                sum += number;
            }

            System.out.println("The average is: " + (sum/n));

        }
    }
```

4. Create a JAVA program which will read n numbers from the keyboard and then will verify all of them if they are perfect squares.
    - Sample Input: n = 5
    - Sample Output:
        - 1 Perfect Square
        - 9 Perfect Square
        - 16 Perfect Square
        - 18 Not a perfect square
        - 25 Perfect Square
    - Solution:

```
    import java.util.Scanner;

    public class Application {

        public static void main(String[] args) {
            Scanner in = new Scanner(System.in);
            System.out.print("How many numbers will you read?  ");
            int n = in.nextInt();

            for(int i = 0; i < n; i++) {
                System.out.print("Enter a number to check if it is a
perfect square: ");
                double number = in.nextDouble();
                double squareRoot = Math.sqrt(number);
                double roundedDownSquareRoot = Math.ceil(squareRoot);
                if((squareRoot - roundedDownSquareRoot) == 0) {
                    System.out.print(number + " Perfect Square");
                } else {
                    System.out.print(number + " Not a perfect square");
                }

            }

        }
    }
```

5. Create a JAVA program wihch will read n numbers from the keyboard and then will display the following pyramid:

```
1
1 2
1 2 3
............
1 2 3 ... n
```

- Solution:

```
import java.util.Scanner;

public class Application {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("N = ");
        int n = in.nextInt();

        for(int i = 1; i <= n; i++) {
            for(int j = 1; j <= i; j++) {
                System.out.print(j + " ");
            }
            System.out.println();

        }

    }
}
```

6. Create a JAVA program which will read two numbers: n and p. The program will display all the powers of n which are smaller than p.
   - Sample Input:
     - N = 2
     - P = 21
   - Sample Output: 0, 1, 2, 3, 4
   - Solution:

```
import java.util.Scanner;

public class Application {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("N = ");
        int n = in.nextInt();
        System.out.print("P = ");
        int p = in.nextInt();
```

```
            for(int power = 0; Math.pow(n, power) <=p; power++) {
                System.out.print(power + " ");
            }

        }
    }
```

7. Write a JAVA program which will display the first N even numbers.
    - Sample Input: 5
    - Sample Output: 0 2 4 6 8
    - Solution:

```
    import java.util.Scanner;

    public class Application {

        public static void main(String[] args) {
            Scanner in = new Scanner(System.in);
            System.out.print("N = ");
            int n = in.nextInt();

            int number = 0;
            for(int i = 0; i < n; i++) {
                System.out.print(number + " ");
                number += 2;
            }

        }
    }
```

8. Create a JAVA program which reads a number `n` from the keyboard and then displays, in descending order, the first odd numbers which are smaller or equal to `n` and greater than 0.
    - Sample Input: 13
    - Sample Output: 13 11 9 7 5 3 1
    - Sample Input2: 12
    - Sample Output2: 11 9 7 5 3 1
    - Solution:

```
import java.util.Scanner;

public class Application {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("N = ");
        int n = in.nextInt();
```

```
            for(int i = n; i> 0; i--) {
                if(i % 2 != 0) {
                    System.out.print(i + " ");
                }
            }

        }
    }
```

Homework exercises

Note: As these are exactly the same exercises since last week, the idea is to use a `for` instruction in order to solve them.

1. Try to imagine what is the result of the following snippets, without using the Eclipse IDE. Afterwards, check it againd the IDE and see if you were right or you need to read the first part again:
    - ```
      int age = 21;
      System.out.println( --score * 12 )
      System.out.println(score++);
      ```
2. Create a JAVA program which will read the integer number `n` from the keyboard and it will display, in descending order, the even numbers which are smaller or equal to `n` but different than 0.
    - Sample Input: 8
    - Sample Output: 8 6 4 2
3. Create a JAVA program which will display the first n odd numbers.
    - Sample Input: 5
    - Sample Output: 1 3 5 7 9
4. Create a JAVA program which will compute the Harmonic Mean of n numbers (n read from the keyboard).
    - Sample Input: 6
        - 1 2 3 4 5 6
    - Sample Output: 2.44
5. Create a JAVA program which will compute `a` to the power `b` without using the `Math.pow` method.
    - Sample Input:
        - a = 3
        - b = 4
    - Sample Output: 81

Guidelines
      ```

- As always, try to resolve the class exercises by yourself and then compare your answers with the provided solution
- Everytime you encounter something new, try to write it down, it will help the process of memorizing it.