Recap to previous lectures!

- How to define method in Java?
- The different types of Java loops? How we define?
- Difference between Break and continue?
- Nested loops?
Lecture 9 : Java Strings

• Java Math.
• Java Strings.
Java Constants

There are several values will never change, e.g. PI is 3.14 and day will always have 24 hours. These values remain constant, will not be modified once assigned.

This is done by adding the keyword modifier final:

```java
final int NUMBER_OF_HOURS_IN_A_DAY = 24;

final float pi = 3.14f;

pi = 5.16f;
```

The same goes for any of the other primitive data type variables.
Java Math

• java.lang.Math is a class that contains methods for performing *basic numeric operations* such as the elementary exponential, logarithm, square root and Generating Random Numbers.

• This section introduces some useful methods in the Math class.
int x = Math.max(3,5);

int y = Math.min(3,5);

double z = Math.max(3.5, 1.2);

int m = Math.abs(-3);                  //absolute value of -3 = 3

double t = Math.pow(2, 4);            //value of 2 to the power of 4 = 16.0

double q = Math.sqrt(16);             //value of square root of 16 = 4.0
Generating Random Numbers

You can use `Math.random()` to obtain a random double value between 0.0 and 1.0, excluding 1.0.

```java
double q = Math.random();
System.out.println("value of random is " + q);

int number1 = (int) (Math.random() * 10);
System.out.println("value of random is " + number1);
```
Generate random within range

\[
\text{Math.random()} \times ((\text{upperbound} - \text{lowerbound}) + 1) + \text{lowerbound};
\]

To generate a number from 10 to 20:

```java
int number1 = (int)(Math.random() * ((20 - 10) + 1) + 10);
System.out.println("value of random is " + number1);
```
How to create a “Rock” “Scissor” “Paper” game ?!
Make a game with random

```
String computerMove;
switch ((int)(3*Math.random())) {
    case 0:
        computerMove = "Rock";
        break;
    case 1:
        computerMove = "Scissors";
        break;
    case 2:
        computerMove = "Paper";
        break;
}
```

```
String computerMove;
int rand;
rand = (int)(3*Math.random());
if ( rand == 0 )
    computerMove = "Rock";
else if ( rand == 1 )
    computerMove = "Scissors";
else
    computerMove = "Paper";
```
Java String

• Sequence of characters e.g.
  String message = "Welcome to Java";

• String is a *predefined class* in the Java with *number of methods*, just like the classes System and Scanner.

• The String type is not a primitive type, known as a reference type means references to an object.
*length()* method to return the number of characters in a string:

```java
String message = "Welcome to Java";
System.out.println("Length of " + message + " is " + message.length());
```

*charAt(index)* method is to retrieve a specific character at certain index.

![Indices and characters diagram](image)
concat() method to concatenate two strings.

```java
String s1 = "Hello";
String s2 = "World";

String s3 = s1.concat(s2); // exactly as s3 = s1 + s2
System.out.println("value of first string is " + s3);

String s4 = s2 + s1;       // exactly as s2.concat(s1)
System.out.println("value of second string " + s4);
```

value of first string is HelloWorld
value of second string WorldHello
BUILD SUCCESSFUL (total time: 0 seconds)
You can still use == and != to compare the contents of two strings

```java
String s1 = "Hello";
String s2 = "Hello";
if(s1 == s2){
    System.out.println("Equal strings !!");
}
else{
    System.out.println("Not equal strings !!");
}
```

Or you can use equals () method

```java
String s1 = "Hello";
String s2 = "Hello";
if(s1.equals(s2)){
    System.out.println("Equal strings !!");
}
else{
    System.out.println("Not equal strings !!");
}
```
**Str.startsWith(prefix)** to check whether string str starts with a specified prefix.

```java
String s1 = "helloworld";
if(s1.startsWith("hello")){
    System.out.println("Got your message !!");
} else{
    System.out.println("incorrect message !!");
}
```

**Str.endsWith(suffix)** to check whether string str ends with a specified suffix.

```java
String s1 = "helloworld";
if(s1.endsWith("world")){
    System.out.println("Got your message !!");
} else{
    System.out.println("incorrect message !!");
}
```
Str.contains(substring) to check whether string str contains string substring.

```java
String s1 = "helloworld";
if(s1.contains("llo")){
    System.out.println("Got your message !!");
}else{
    System.out.println("incorrect message !!");
}
```
You can obtain a single character from a string using the `charAt` method. You can also obtain a substring from a string using the `substring` method in the `String` class:

```java
String s1 = "helloworld";
String s2 = s1.substring(2,5);
System.out.println(s2);
```

```
ll
BUILD SUCCESSFUL (total time: 0 seconds)
```