



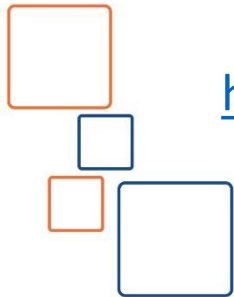
ITI Java DevBelt: **Yellow Belt**

CORE JAVA PROGRAMMING

AN INTRODUCTION TO JAVA

Course link on maharatech:

<https://maharatech.gov.eg/course/view.php?id=2052>



Java™ Education
and Technology Services

Invest In Yourself,
Develop Your Career

Week 1

CH01 - The History and Evolution of Java

CH02 - An Overview of Java

CH03 - Datatypes, Variables and Arrays

CH04 - Operators, Control Statements and String Handling



Ch2 : Lab Exercises

- Create simple applications that performs the following actions:
 - Display a message to the command prompt.
 - Receives an input and checks for its value and prints it back.
 - Receives two inputs a number and a string and prints the string on different lines according to the given number.

Ch3: Lab Exercise

- Develop an application that extracts the minimum and maximum of the elements of an array of 1000 element and compute the search running time.
- Rewrite the application to implement the binary search algorithm and compute the search running time.

Hint: Use `System.currentTimeMillis()` or `System.nanoTime()`.

Ch4: Lab Exercise

- Given a sentence and a word, your task is that to count the number of occurrences of the given word in the string and print the number of occurrence of the word.
 1. Perform the above task using only methods of the String class (2 ways).
 2. Perform the above task using StringTokenizer class
- Count a group of words in a string using regular expressions

Week 2

CH05 - Packages and Interfaces

CH06 - Wrapper Classes, Enumeration, Autoboxing and Annotations



Ch5 : Lab Exercises

- Develop an application to convert temperature from Centigrade to Fahrenheit using `Function<T,R>`
- Use the interfaces in `java.util.function` to build an application that defines the roots of the quadratic equation ($ax^2+bx+c=0$) and the roots could be computed by the following formula ($x=(-b\pm\sqrt{b^2-4ac})/2a$)

Ch6 : Lab Exercises

- Create a custom annotation named **Author** and allow it to be used on the class, method, constructor, and member level.
- Create a class annotated with **@Author**
- Using reflection to extract information regarding the annotated members

Week 3

CH07 - Exception Handling

CH08 - Generics and Lambda Expressions



Ch7 : Lab Exercises

- Create your own exception class and write down two other classes, the first will contain three methods throwing your newly created exception class and the second class will be calling the methods that throws exception using the try-catch-finally block.

Ch8 : Lab Exercises 1

- Create a base class named **Shape** that contains one abstract method `draw()`.
- Create two concrete classes (**Rectangle** and **Circle**) that extend Shape
- Create a test class that defines a method that accepts a list that contains only child classes of shape
- Test your method by creating two **ArrayList** of Rectangle and shapes and pass them to the generic method

Ch8 : Lab Exercises 2

- Create a generic class that could be used to represent complex numbers
- Create some generic methods that represent basic arithmetic operation on complex(addition, subtraction, etc...)

Ch8 : Lab Exercises 3

- Your goal is to make a method called `betterString` that takes two `Strings` and a lambda that says whether the first of the two is “better”.
- The method should return that better `String`; i.e., if the function given by the lambda returns `true`, the `betterString` method should return the first `String`, otherwise `betterString` should return the second `String`.
 - `String string1 = ...;`
 - `String string2 = ...;`
 - `String longer = StringUtils.betterString(string1, string2, (s1, s2) -> s1.length() > s2.length());`
 - `String first = StringUtils.betterString(string1, string2, (s1, s2) -> true);`

Ch8 : Lab Exercises 4

- Given a String, the task is to check whether a string contains only alphabets or not.
- use `isLetter()` method of `Character` class.

Week 4

CH09 - Java Collections

CH10 - Java Stream API

Next Step

CH11 - Threads



Ch9 : Lab Exercises

- Simple Word Dictionary
 - Develop an application that stores words in an array into a collection.
 - Create a map that uses the alphabets as keys and a collection as values (This collection should only contain words starting with the corresponding key)
 - Elements in the words map for each alphabet should be sorted
 - Provide methods to print all the letters and corresponding words
 - Provide a method to print the words of a given letter

Ch10 : Lab Exercises

- World Countries: There are two domain classes: Country and City. Each city belongs to a country defined by the attribute, countryCode. Each country has a unique code and has many cities.
 - The Country class has the following attributes
 - The City class has the following attributes
- ```
private int id;
private String name;
private int population;
private String countryCode;
```
- ```
private String code;  
private String name;  
private String continent;  
private double surfaceArea;  
private int population;  
private double gnp;  
private int capital;  
private List<City> cities;
```
- Create the above two classes and then create a java application to find the following:
 - Find the highest populated city of each country
 - Find the most populated city of each continent
 - Find the highest populated capital city