

Formation continue informatique

Programmation - Java - Initiation

Overview of the course

Introduce students to the Java Language Programming.

Explain to students the fundamental concepts of Java.

Explain to students the basics of Java

Explain to students the control flow statements of Java

Explain to students how to manage the exceptions that the Java can launch at running time.

Explain to students how load and create file.

Show the students how to create a modular Java project using also 3° part API.

Structure of the course

The course is structured in 8 self-contained modules. Each module is supposed to require from 2 to 8 hours and it contains the theoretic part followed by some exercises to test if the topic has been understood

Organization of each lesson

Theoretic part with demonstration of the arguments explained

Practical part where the students are demanded to implement what has been explained to them

Maximum interaction during the lectures it well appreciated

Outcome of the course

Create a project that comprehends the topics touched in all modules

Optional

Homework

Module 1: Discover the JAVA programming Language

Java History – The reason why it was invented

Illustrate the different Java Editions

Explain the Java Platform

- Show a really simple code example

Formation continue informatique

- Explain the structure of a java project
- Show how to install the JVM
- Explain how to compile it from DOS
- Explain how to execute it from DOS
- Explain how to create an executable (JAR) file and run it on DOS

Overview of Java Development Frameworks

- Download and install the Eclipse framework
- Explain how to create a java program on Eclipse
- Explain how to run a java program on Eclipse
- Explain how to create an executable (JAR) file from Eclipse
- Explain the structure of an Eclipse Java project

Introduction to Java API

Exercise:

Discussion about the Java Language Programming

Formation continue informatique

Module 2: Object-Oriented Programming Basic Concepts

Classes and Objects Theory

- What is a Class
- What is an Object

Variables

- What is a variable
- How to declare and instantiate a variable

Methods

- What is a Method
- How to use Methods

Classes & Object Practice

- Declaring a Class
- Creating an Object
- Declaring Member Variables

Comments

- Why to comment code
- How to comment code
- How to create documentation

Exercise:

Compile and execute code from DOS

Create a simple java program and run it on eclipse

Formation continue informatique

Module 3: Variables, Constants, Operators and Control Flow Statements

Variables

- Primitive Data Types
- Object Variables: Array & HashMap Classes

Constants

Operators on Variables

Operators on Objects

Control Flow Statements

- if-then-else
- for
- while
- switch-case

Exercise:

Create and Manipulate Variables.

Implement a small algorithm using all the Control Flow Statements.

Module 4: File Manipulation & Stream

Java.io API

How to Load and Save Files

Serialization and Deserialization

Binary File

XML File

JSON File

Exercise:

Implement a small application that loads a .txt file from the disk, and serializes it in JSON and XML format

Formation continue informatique

Module 5: Modeling a Java Program – Interfaces and Inheritance and UML Diagram Class

Explain What Is an Interface?

- Defining an Interface
- Implementing an Interface
- Using an Interface as a Type
- Evolving Interfaces

Explain What Is Inheritance?

- Multiple Inheritance of State, Implementation, and Type
- Overriding and Hiding Methods
- Polymorphism
- Hiding Fields
- Using the Keyword super
- Object as a Superclass
- Writing Final Classes and Methods
- Abstract Methods and Classes

UML Class Diagram

- Introduce UML and UML Tools
- Illustrate a UML Class Diagram
- Show how to Model a Simple Class Diagram

Exercise: students will create an application structured as a diagram UML that I will provide them

Formation continue informatique

Module 6: Object-Oriented Programming Advanced Concepts

This Keyword

Controlling Access to Member of a Class

Class Member

Nested Classes

- Inner Classes
- Local Classes
- Anonymous Classes

Exercise:

Implement the same code using the different Nested Classes

Module 7: Generics

Why Use Generics?

Generic Types

Generic Methods

Bounded Type Parameters

Generic Methods and Bounded Type Parameters

Exercise:

Generics in Action

Formation continue informatique

Module 8: Program Configuration, Logging and Tests

How to configure externally parameters of a java program

External Java API

Introduction to external Java API

How to understand and use external Java aPI

How to manage the Logging of a program

Simple Logging Java Facade

How to do Unit Tests (Optional)

JUnit

Exercise:

Implement an application that requires the modification of the Build Path