Course Name:	**Programming languages & software development**
Duration:	6 **months**
Level:	**Diploma**
Mode of Learning:	**Online**
Prerequisite:	**Mature Entry (21 years) with 3 years of related work experience, 5 CXC certificates (Math and English mandatory), Any certifications relevant to subject matter**

Abstract

The programming languages and software development course is designed to provide students with a deep understanding of programming languages, software development concepts, and best practices. The course typically covers a range of programming languages and development tools and may include hands-on coding assignments and projects.

Course Content

Performance-Based Assessment Task: Performance-Based Task is an application-oriented end-of-the-course assessment. This Hands-on project will encourage the learners to apply the knowledge gained throughout the programme and build practical skills.

- Programming Languages
- Software Development
- Development Tools
- Web Development

Certification

THE DUAL CERTIFICATE is awarded by EIU-Paris (University) and Pedagog

Please Note: A Hard Copy of the certificate from the university can be requested at an additional minimal fee, via e-mail to hello@pedagog.ac

For more information mail hello@pedagog.ac

LESSONS COVERED:

⌨ (00:05) Introduction
⌨ (01:37) What is Programming?
⌨ (06:19) How do we write Code?
⌨ (11:44) How do we get Information from Computers?
⌨ (14:46) What can Computers Do?
⌨ (20:43) What are Variables?
⌨ (25:02) How do we Manipulate Variables?
⌨ (31:54) What are Conditional Statements?
⌨ (37:54) What are Array's?
⌨ (44:26) What are Loops?
⌨ (49:37) What are Errors?
⌨ (55:22) How do we Debug Code?
⌨ (1:00:25) What are Functions?
⌨ (1:09:52) How can we Import Functions?
⌨ (1:13:45) How do we make our own Functions?
⌨ (1:21:56) What are ArrayLists and Dictionaries?
⌨ (1:27:38) How can we use Data Structures?
⌨ (1:36:27) What is Recursion?
⌨ (1:43:42) What is Pseudocode?
⌨ (1:50:40) Choosing the Right Language?
⌨ (1:55:34) Applications of Programming

Software Engineering Principles Lecture 01: The Software Crisis
Software Engineering Principles Lecture 02: Objects and Information Hiding
Software Engineering Principles Lecture 03: Responsibilities
Software Engineering Principles Lecture 04: Class Relationships and Collaborations
Software Engineering Principles Lecture 05: Class Hierarchies
Software Engineering Principles Lecture 06: Subsystems
Software Engineering Principles Lecture 07: Protocols
Software Engineering: Unit and Integration Testing
Software Engineering: A Case Study with a Real Example - CSV 2 XML
Software Engineering Principles: The Software Crisis
Software Engineering Principles Revision Lecture: the 1st 30 minutes
A Visual Interface for Complete Software Testing
Setting Up Git and GitLab with Netbeans on Mac OS X
Hello Netbeans World: Setting Up Netbeans 12 on Mac OS X - Installation and New Projec
Source Code Documentation and Version Control
Setting Up Netbeans and Version Control (Git + GitLab) On Mac OS X
Object-Oriented Design: Objects and Responsibilities (Part I of 2)
Object-Oriented Design: Objects and Responsibilities (Part 2 of 2)
Object-Oriented Design: Collaborations
Bob's Concise Coding Conventions (C3)
Object-Oriented Design: Hierarchies
Object-Oriented Design: Subsystems
Object-Oriented Design: Protocols
Guidelines on Debugging Source Code
Software Engineering with Design Patterns, Part 1 of 2
Software Engineering Design Patterns (Part 2 of 2)
Developing Maintainable Software Exam Revision
Hello IntelliJ World: Setting up IntelliJ on the Mac OS (From Scratch)
Setting Up a Maven Project in IntelliJ (From Scratch)
Setting Up a Gradle Project in IntelliJ (From Scratch)

Object-Oriented Design: Objects and Responsibilities (Condensed Version)
Object-Oriented Design: Collaborations and Hierarchies (Part 1 of 2)
Object-Oriented Design: Hiearchies Part 2 and Subsystems Part 1
Object-Oriented Design: Subsystems Part 2 and Protocols Part 1
Object-Oriented Design: Protocols Part 2 and Design Patterns Part 1
Software Design: Design Patterns Part 2
Setting Up a JavaFX Project in IntelliJ (from scratch)
Object-Oriented Design: Pop Quiz on Identifying Objects Exercise + Introduction to Maven
Setting Up Unit Testing in a JavaFX Project Using IntelliJ (from Scratch)
Open Source Software and Software Libraries
Lecture on Software Licenses and Libraries
Object-Oriented Design Exercise: Identifying Responsibilities
Guest Lecture from Industry: Curiosity Software Ltd on Automated Software Testing
Tales from the Software Trenches: Some of Bob's Unbelievable Stories from Industry
Advanced Visual Debugging Strategies
Exam Revision for Developing Maintainable Software
Object-Oriented Design: Subsystems
Object-Oriented Design: Collaborations and Hierarchies
Setting Up JUnit for a JavaFX Project in IntelliJ from Scratch (Screencapture demo with Voiceover)