

# JAVA PROGRAMMING

## Tutorial 07

### Activity 01: Student Management System

In this exercise, you will design and implement a simple student management system. The system needs to collect and process data about **students** and **courses**. A student contains **student ID**, **name**, **date of birth**, **GPA** (grade point average), and **a list of enroll courses**. A course contains data on **course ID**, **name**, **description**, **credits**, and **academic transcript**. The following is a partially completed table of the domain constraints that apply to the attributes of the two classes.

Class	Attribute	type	mutable	optional	length	min	max
Student	studentID	Integer	F	F	-	1	-
	name	String	F	F	50	-	-
	dateOfBirth	String	F	F	10	-	-
	gpa	Double	T	T	-	0.0	10.0
	enrolledCourses	List	T	T	-	-	-
Course	course ID	Integer					
	name	String					
	description	String					
	credit	Integer					
	academicTranscript	List					

Complete the following tasks:

- Complete the domain constraints in the table, using your practical understanding of the application.
- Create a Java package called “**tut07.lms**” which contains all implemented classes in this exercise.
- Determine a minimum set of operations needed for each class. Justify your choice of each operation. Draw a UML class diagram.
- Review the design and implement all the classes.
- Create a test program named **LearningManagementSystem** to ensure everything is working properly.

## Activity 02: Library management

**Library Management System** is software built to handle the primary housekeeping functions of a library. Libraries rely on library management systems to manage asset collections as well as relationships with their members. Library management systems help libraries keep track of the books and their checkouts, as well as members' subscriptions and profiles. In this exercise, your task is to create a simple system using Java. Here are the main classes of the system:

- ✚ **Account:** We will have two types of accounts in the system, one will be a general **member**, and the other will be a **librarian**. An account contains data **about ID, Password, Status** (Active, Disabled, Restricted, Blacklisted, Deleted) and **Person**.
- ✚ **Book:** The basic building block of the system. Every book will have **ISBN, Title, Subject, Authors, Publishers, Language, number of pages**, etc.
- ✚ **BookItem:** Any book can have multiple copies; each copy will be considered a book item in our system. Each book item will have a unique **Barcode, Price, Format** (Hardcover, Paperback, Audiobook, Ebook, Newspaper, Magazine, Journal), **Status** (Available, Reserved, Loaned, Lost), **Date of Purchase, Publication Date**, etc.
- ✚ **Catalog:** Catalogs contain a list of books sorted on certain criteria. The catalog will serve as a centralized repository, housing lists of books categorized based on various criteria, including **Title, Author, Subject**, and **Format**. To enable seamless searching, we will override abstract methods from an interface named **Search**. This interface will define the blueprint for search functionality across different catalog criteria, ensuring uniformity and flexibility. Each catalog entry will contain essential information such as **the creation date, total number of books**, and **a mapped list of book titles, authors, subjects, and formats**.
- ✚ **BookLending:** Responsible for managing reservations and the checking-out of book items. This class will have at least six attributes: **Creation date, Account, Book items, Due date, Return date, Status**

(Available, On Loan, Reserved, Overdue, Renewed, Recalled, Lost, Damaged, In Transit, On Hold, Cancelled, Not Available).

Complete the following tasks:

- ❖ Create a package named “**tut07.library**” and:
  - + Create a completed table of the domain constraints that apply to the attributes of the classes.
  - + Determine the minimum set of attributes/operations required for each class. Please provide a justification for the selection.
  - + Draw a UML class diagram.
  - + Review the design and implement all the classes.
  - + Create a test program to ensure everything is working properly.

## Submission

Submit a zip file containing all Java programs to this tutorial's submission box in the course website on FIT Portal.