EE3206: JAVA PROGRAMMING AND APPLICATIONS

Effective Term
Semester A 2025/26

Part I Course Overview

Course Title
Java Programming and Applications

Subject Code
EE - Electrical Engineering
Course Number
3206

Academic Unit
Electrical Engineering (EE)

College/School
College of Engineering (EG)

Course Duration
One Semester

Credit Units
3

Level
B1, B2, B3, B4 - Bachelor's Degree

Medium of Instruction
English

Medium of Assessment
English

Prerequisites
EE2331 Data Structures and Algorithms
or equivalent

Precursors
Nil

Equivalent Courses
EE2311 Object-oriented Programming and Design

Exclusive Courses
Nil

Part II Course Details

Abstract

The aim of this course is to provide students with an understanding of the object-oriented design and programming techniques. Java, a prime object-oriented programming language, is used to illustrate this programming paradigm.

Course Intended Learning Outcomes (CILOs)

	CILOs	Weighting (if app.)	DEC-A1	DEC-A2	DEC-A3
1	Apply object-oriented programming paradigm in software design		x	x	
2	Implement Java code following standard coding conventions used in the software community.		x	x	
3	Apply the built-in data structures and collections framework of Java to solve more complex computation problems.		x	x	x
4	Apply the Java Stream API and Functional interfaces to implement data processing software using the functional programming approach.		x		
5	Apply system-level techniques supported in Java to implement event-driven and multi-thread software that can take advantage of multi-cores CPU available in today's computer systems.		x	x	x

A1: Attitude
Develop an attitude of discovery/innovation/creativity, as demonstrated by students possessing a strong sense of curiosity, asking questions actively, challenging assumptions or engaging in inquiry together with teachers.

A2: Ability
Develop the ability/skill needed to discover/innovate/create, as demonstrated by students possessing critical thinking skills to assess ideas, acquiring research skills, synthesizing knowledge across disciplines or applying academic knowledge to real-life problems.

A3: Accomplishments
Demonstrate accomplishment of discovery/innovation/creativity through producing /constructing creative works/new artefacts, effective solutions to real-life problems or new processes.

Learning and Teaching Activities (LTAs)

	LTAs	Brief Description	CILO No.	Hours/week (if applicable)
1	Lecture and Tutorial	Teaching activities are primarily based on lectures followed by examples to show students the basic skills. Tutorials are conducted in the laboratory. Students will acquire the programming skills via hands-on experiences in writing Java codes.	1, 3, 4, 5	3 hrs Lect/wk 1 hrs Tut/wk

2	Assignments	AAssignments are designed to help students to acquire the essential programming skills, and get familiar with the software development tools through hands-on coding practices.	1, 3, 4, 5	2 to 3 hrs/wk

Assessment Tasks / Activities (ATs)

	ATs	CILO No.	Weighting (%)	Remarks ("-" for nil entry)	Allow Use of GenAI?
1	Tests	1, 3, 4, 5	25	-	No
2	#Assignments (3 to 4)	1, 3, 4, 5	25	-	Yes

Continuous Assessment (%)

50

Examination (%)

50

Examination Duration (Hours)

2.5

Minimum Continuous Assessment Passing Requirement (%)

30

Minimum Examination Passing Requirement (%)

30

Additional Information for ATs

To pass the course, students are required to achieve at least 30% in course work and 30% in the examination.

Assessment Rubrics (AR)

Assessment Task

Examination

Criterion

Achievements in CILOs

Excellent (A+, A, A-)

High

Good (B+, B, B-)

Significant

Fair (C+, C, C-)

Moderate

Marginal (D)

Basic

Failure (F)
Not even reaching marginal levels

Assessment Task
Coursework

Criterion
Achievements in CILOs

Excellent (A+, A, A-)
High

Good (B+, B, B-)
Significant

Fair (C+, C, C-)
Moderate

Marginal (D)
Basic

Failure (F)
Not even reaching marginal levels

Part III Other Information

Keyword Syllabus

Object-oriented principles and design
Objects and classes; information hiding; encapsulation; data abstraction; inheritance and polymorphism; unified modeling language (UML) and diagrams.

Overiew of the Java language
Basic syntax; Classes, interfaces, and methods; Fundamental data types and wrapper classes; Mutable and immutable objects; String and related classes; Class design; Generic classes and methods; Basic I/O processing; Exception handling; Java coding conventions and guidelines.

GUI programming
Graphical user interface components; frame and layout management; event-driven programming and action handler.

Software design using Java
Sorting and searching; Java classes and interfaces that support commonly used data structures such as list, stack, queue, tree, map, and hash table. Applications of the Java Collection Framework in conventional imperative programming paradigm.

Functional/declarative programming paradigm for data processing applications; Java Functional interfaces; Lambda expression; Java Stream API; Stream data pipeline.

Multi-thread software design; Thread life cycle; Thread safety and management; Thread-safe class design; Data sharing, synchronization and mutual exclusion; Re-entrance lock and condition objects; Parallel processing and the Java Fork/Join processing framework.

Reading List

Compulsory Readings

	Title
1	Cay Horstmann, Big Java, latest edition

Additional Readings

	Title
1	K. Sharan, Beginning Java 8 Language Features, Lambda Expression, Inner Classes, Threads, I/O, Collections and Streams, Apress 2015
2	https://docs.oracle.com/en/java/javase/21/docs/api/index.html http://docs.oracle.com/javase/tutorial/