# COURSE STRUCTURE

**Name of Course: OBJECT-ORIENTED PROGRAMMING**

**Course Code:** ICT2113

**Credit Hours: 4**

**Prerequisite/co-requisite: ICT1103 Structured Programming**

**Summary:** This course exposes students to the concepts of object-oriented programming. It focuses on the Java programming language. It covers data types, operators, control statements, methods, classes and objects, inheritance and polymorphism, abstraction and interfaces.

**Course Learning Outcomes:**

Upon completing this course, the students will be able to:

CLO1: Explain the concepts of object-oriented programming (C2,PLO1)
CLO2: Display the ability to utilize classes in enhancing Java applications (P4,PLO6)
CLO3: Display the ability to use packages, interfaces, I/O, exceptions, string objects and data objects in Java (P4, PLO3)
CLO4: Display the ability to execute and run a Java technology application (P3, PLO3)

**Course Format:**

| **Total Student Learning Time (SLT)** (L = Lecture; T = Tutorial; P = Practical; EL = E-Learning)**:** | | | | | |
|---|---|---|---|---|---|
| **Learning Hours** | | | | **Independent Learning (hr)** | **Total Student Learning Time (hr)** |
| L | T | P | EL | | |
| 28 | 0 | 26 | 14 | 92 | 160 |

**Teaching and Delivery Methods/ Teaching Methodology:**

Lectures, Tutorial and Practical/Laboratory work delivered in a combination of blended & independent learning

E-Learning provided by INTI makes learning more accessible and convenient for the students. The blended model utilized by INTI is the integration of E-learning via INTI's Learning Management System and the conventional lecturer-led classroom activities. INTI students are required to access to the online learning materials (additional notes, reading materials, online assessments, discussion forums and etc.), so as to acquire a complete learning process. This also promotes self-directed learning in encouraging INTI students to be independent learners.

**Syllabus**:

| Course Content Outline | CLO* |
|---|---|
| **Introduction to Java Programming -** Key features of Java technology, Compile and execute a simple Java application, Display text, Escape sequences, Using comments, Data types, variables, and constant, Arithmetic operators, Input/Output using JOptionPane Dialog Boxes | 1,2,4 |
| **Expressions and Flow Control -** Type casting, Equality and relational operators, Logical operators, Selection control structures, Repetition control structures, Break and continue statements, Labeled break and continue statements | 1, 2,4 |
| **Fundamental of Classes -** Class Basics, variables, methods and objects definition, Creating and using instance variables / methods and class variables / methods, Creating object, Constructors, Using object references, Accessor and Mutator methods, The object reference this, static variables, static methods, Definition on package and import statements, Creating packages | 1,2,4 |
| **Fundamental of Methods -** Methods of class Math: pow, sqrt, round, min, max, random, abs, ceil, and floor,Methods of class String: length, toUpperCase and toLowerCase, charAt, getChars, valueOf, equals, equalsIgnoreCase, compareTo, The toString methods, Recursive methods | 1,2,4 |
| **Arrays -** Single-dimensional arrays: Declare and create arrays of primitive, class, or array types, initializing arrays, Accessing array elements, Copying array, and Passing arrays to methods,<br>Sorting (Bubble Sort) and Searching (Linear Search and Binary Search) arrays, Multidimensional array: two-dimensional and three-dimensional | 1,2,4 |
| **Object-Oriented Programming -** Object oriented concepts and modeling (encapsulation, inheritance and polymorphism), public and private access modifiers, Composition, Overloading and overriding methods | 1,2,3,4 |
| **Advanced Class/Interface Features -** Create final variables, Enumeration types, Abstract classes and methods, Concrete classes, Interfaces | 1, 2, 3,4 |
| **Exceptions -** Define exceptions and its categories, Identify common exceptions, Use try, catch and finally blocks, Catching multiple exceptions | 2, 3,4 |

**Student Evaluation:**

| **Continuous Assessment** | | **Percentage (%)** | **CLO** |
|---|---|---|---|
| 1 | Test | 20 | 1 |
| 2 | Lab tutorial | 10 | 2 |
| 3 | Assignment | 10 | 3 |
| 4 | Project | 20 | 4 |
| **Final Assessment** | | **Percentage (%)** | **CLO** |
| Final Exam | | 40 | 1 |
| **Total** | | **100%** | |

**Final exam format**:
Duration: 2 hours
The students will be required to answer:
Section A: **Answer All**
Section B: **Answer All**

**Grading Scale:**
A+ (90-100), A (80–89), A- (75-79), B+ (70-74), B (65–69), B- (60–64), C+ (55–59) C (50–54), C- (45–49), D (40–44), F (0–39), RP (Resit Pass) Marks capped at 50, RF (Resit Fail) (0-49)

**IMPORTANT NOTE:**
STUDENTS ARE REQUIRED TO **"PASS"** BOTH CONTINUOUS AND FINAL ASSESSMENT IN ORDER TO PASS THE SUBJECT

**Additional Information:**

**Main Reference(s) Supporting Course:**

1. Y. Daniel Liang (2019), Introduction to Java Programming and Data Structures, Comprehensive Version, 12th Edition, Pearson, ISBN: 978-0136520238

2. Tony Gaddis (2019), Starting Out with Java: From Control Structures through Objects, 6th Edition, Pearson, ISBN: 978-0133957051

**Additional References:**

1. Lewis J. and Loftus W., (2018), Java: Software Solutions: Foundations of Program Design, 9th Edition, Pearson Education. ISBN-13: 9780134543284

**LABORATORY WORK:**

| Week | Practical Work |
|---|---|
| 1 | Compiling and Running a Program |
| 2 | Flow Control: selection |
| 3 | Flow Control: repetition |
| 4-5 | Creating Classes, Constructor, Mutators and Accessors |
| 6 | String Manipulation |
| 7-8 | Methods |
| 9-10 | Array |
| 11-13 | Inheritance, Polymorphism, Abstract Classes and Methods |