A. First Part - Programming basics with Java

1. Introduction / Software Setup
    a. Introduction to ICT (Career plan)
        - Syllabus introduction & course procedures
        - Requirements
        - Basics of ICT
        - Information in digital form, number systems, binary system
    b. What is programming?
        - Introduction to Programming
        - Algorithmic thinking, reasoning
    c. What are Programming Languages (PLs)?
    d. How to choose a PL to learn?
    e. JDK, JRE, JVM, IDE?
        - Platform independency, C++ vs Java
        - javac vs java
        - How to compile & run java code from terminal/cmd
        - .java and .class files - source code & bytecode & machine code
        - Compiling vs interpretation vs running
        - JDK & JRE & JVM?
        - IDEs - Intellij IDEA, NetBeans, Eclipse
    f. What is VCS (Git / GitHub)?
        - Git download and installing
        - Overview about version control systems
        - Initializing or cloning a repository
        - Basic git commands: clone, status, add, commit, push, pull
2. Java basics
    a. Java syntax, writing first “Hello, World!” app in Java
    b. Manifest: public static void main (String[] args) { … }
    c. Print to console
        - System.out.print(“Hello, World”);
        - System.out.println(“Hello, World”);
        - System.out.printf(“Hello, %s”, “World”);
        - System.out.printf(“Hello, World: %.2f”, 50);
    d. Storing data - Variables - declaration & initialization
    e. Data types
        - Primitive types
            a. byte, short, int, long, float, double,
            b. char, boolean
        - Reference types

   f. Comments
      - Single line comment
      - Multiple lines (block) comment
      - Documentation comment
   g. Operations
      - Arithmetic operations
      - Relational operations
      - Logical operations
      - Assignment operations
      - Miscellaneous operations
3. Control Flow
   a. Input from console - Scanner class
   b. Code structure: input -> process -> output
   c. Conditional statements
      - if
      - if - else
      - if - else if - else
      - switch - case
      - Ternary operator
   d. Loops
      - for
      - while
      - do-while
      - break, continue
   e. Nested conditions and loops
4. Arrays
   a. Declaration, initialization of arrays
   b. Operations on an array (fill, print, find max, min, copy etc.)
   c. Enhanced for loop (“for-each”)
   d. How memory works for arrays (stack vs heap memory)
   e. Two and more dimensional arrays
5. Methods
   a. Declaration of methods, method signature
   b. Parametric & non-parametric methods
   c. Void & value methods
   d. Overloading, rules for overloading
6. First exam - Fundamentals of Programming

7. Object-Oriented Programming (OOP) - #1
    a. Object and class
    b. Constructors, object initialization
    c. Types of variables
        - Instance variables
        - Local variables
        - Static (global) variables
    d. Static vs non-static methods and variables
    e. References/Garbage Collection
    f. Getters and setters
8. Object-Oriented Programming (OOP) - #2
    a. Encapsulation
    b. Inheritance
    c. Polymorphism
    d. Abstraction
    e. Keywords: this & super & instanceof
    f. @Override
    g. Compile-time (overloading) vs runtime (overriding) polymorphism
9. Object-Oriented programming (OOP) #3
    a. Abstract classes
    b. Interfaces
    c. Abstract classes vs interfaces
    d. Functional & Marker Interfaces
10. Object-Oriented programming (OOP) #4
    a. Enumeration
    b. Immutability
        - Final class
        - Final method
        - Final field, params
    c. Var keyword
11. Object-Oriented programming (OOP) #5
    a. Packaging, built-in packages
    b. Importing: single vs whole imports (wildechart)
    c. Static imports
    d. UML diagrams for class designing
    e. Wrapper types
    f. Casting (upcating, downcasting)
    g. Boxing and unboxing. Autoboxing

12. String class
    a. Character array and understanding String
    b. String under the hood
    c. Methods of String class (some)
        - toLowerCase() & toUpperCase()
        - substring() & trim()
        - indexOf(String s) & indexOf(int i)
        - split(), replace(), length(), concat()
    d. Memory (RAM) intro (stack vs heap)
    e. Memory for String management, String pool
    f. Reference and how this works?
    g. Passing values
        - Passing-by-value
        - Passing-by-reference
    h. String concatenation:
        - “+” operator for strings
        - concat()
        - StringBuilder
        - StringBuffer
        - Comparison of above solutions

13. Second exam - Object Oriented Programming

14. Exceptions
    a. Exception hierarchy
    b. Error vs Exception
    c. Checked and unchecked exceptions
    d. Try-catch
    e. Multiple catch and union catch
    f. Swallowing exceptions
    g. Try-with-finally (in files)
    h. Try-with-resources (in files)
    i. Custom Exceptions
    j. throw vs throws
15. Date and Time API
    a. LocalDate
    b. LocalTime
    c. LocalDateTime
    d. Date vs LocalDate
    e. java.util.Date vs java.sql.Date

16. Generics & Optional
    a. Need for Generics
    b. Diamond operator
    c. Type wildcards (lower and upper bounds)
    d. Generic class definitions
    e. Generic method definitions
    f. Optional class and its usage
    g. Introduction to Functional Programming
    h. Method chaining strategy
17. Sorting and Comparing
    a. Comparable vs Comparator
    b. Functional Interfaces
    c. Lambda expressions
    d. Method references
    e. Arrays.sort()
18. Introduction to Algorithms
    a. Introduction to complexity analysis
    b. Notations
        - Worth case scenario (Big O)
        - Best case scenario (Big Omega)
    c. Searching
        - Linear search
        - Binary search
    d. Sorting
        - Bubble sort
        - Selection sort
        - Merge sort
19. Introduction to Data Structures
    a. Collections class
    b. Introduction to Java Collection Framework (API)
        - ArrayList
        - LinkedList
        - Map, hashing
        - Set
        - Vector
        - Stack
        - Queue
        - Deque
        - HashSet vs LinkedHashSet vs TreeSet
        - HashMap vs LinkedHashMap vs TreeMap

20. Introduction to Java Stream API
    a. Input -> Process -> Output
    b. Source -> Intermediate -> Terminal operations
21. File Input/Output
    a. File reading and writing with “io”
    b. Input, output, error with System class (in, out, err)
    c. Character streams vs byte streams
    d. FileReader and FileWriter
    e. Buffered file operations
    f. File reading and writing with “nio”
22. Serialization, Reflection
    a. Serialization, object streams
    b. Writing object into file (text vs object)
    c. Binary vs XML vs JSON serialization
    d. Transient keyword and its mechanism
    e. Introduction to Reflection API
    f. Java class object, fields, methods, constructors
    g. Dynamic invocation, annotations
23. Multithreading
    a. Introduction to multithreading, process vs thread vs task
    b. Thread class
    c. Runnable interface
    d. Callable interface
    e. Execution service
    f. Concurrency API
    g. Atomic Scalars
24. Creating proper project structure
    a. Input -> process -> output
    b. Controller -> Service -> DAO
    c. Protocoling
    d. Maven, packaging

25. Third exam - Java SE - MODULE #1