Module 1 Assignment – Java Advanced Concepts & Core Foundations

Deadline: 25/10/2025

Total Marks: 100 (75 Marks - Coding, 25 Marks - Theory)

SECTION A – Theory Questions (25 Marks)

1. Java I/O Streams (5 Marks)

- Explain the difference between Byte Stream and Character Stream in Java.
- Give one example of each.

2. Collection Framework (5 Marks)

- Compare ArrayList and LinkedList.
- When would you use one over the other?

3. Lambda Expressions (5 Marks)

- Explain what Lambda Expressions are and how they improve code readability.
- Give a simple example of using a lambda expression with a functional interface.

4. Generics and Annotations (5 Marks)

- What are Java Generics? Why are they important for type safety?
- Also, explain the role of annotations in Java with examples.

5. JDBC (5 Marks)

- Explain the steps required to connect a Java application to a database using JDBC.

SECTION B – Coding Problems (75 Marks)

All problems must be solved in Java 17 or above. Use proper naming conventions, exception handling, and JavaDoc comments where appropriate.

Problem 1: File Word Counter (Java I/O) – [10 Marks]

Problem Description

Write a program that reads a text file and prints the number of words, characters, and lines present in it.

Example

```
Input File Content:
Java is fun
Learning Java I/O

Output:
Words: 5
Characters: 28
Lines: 2
```

Constraints

- Use FileReader or BufferedReader
- Handle file not found gracefully

Problem 2: Student Record Manager (Collection Framework + Generics) – [15 Marks]

Problem Description

Design a Student class with fields id, name, and marks. Store multiple students using a Generic Collection and perform:
- Add students
- Sort students by marks (descending)
- Display top 3 students.

Example

```
Input:
(1, Alice, 85)
(2, Bob, 91)
(3, Charlie, 78)
(4, David, 92)

Output:
Top 3 Students:
David (92)
Bob (91)
Alice (85)
```

Constraints

- Use Generics
- Use Comparator or Lambda Expressions for sorting

Problem 3: String Filter Using Lambda (Lambda Expression) – [10 Marks]

Problem Description

Given a list of strings, use Lambda Expressions and the Stream API to:
- Filter out strings shorter than 5 characters
- Convert remaining strings to uppercase
- Print the result

Example

Input:

```
[java, lambda, code, expression, api]
```

Output:

```
[LAMBDA, EXPRESSION]
```

Constraints

- Use Stream API
- Use Lambda Expression

Problem 4: Logging with Custom Annotation (Annotation + Reflection) – [15 Marks]

Problem Description

Create a custom annotation @LogExecutionTime. Apply it to a method and when executed, log how long it took to execute using reflection.

Example

Input:

```
@LogExecutionTime
public void processData() {
   Thread.sleep(2000);
}
```

Output:

```
Method processData executed in 2005 ms
```

Constraints

- Use Reflection API
- Use RetentionPolicy.RUNTIME

Problem 5: Database CRUD Operations (JDBC) – [25 Marks]

Problem Description

Create a small Java application to perform CRUD operations on a student table in MySQL using JDBC.

Table: id(INT), name(VARCHAR), marks(INT)
Operations: Insert, Read, Update, Delete

Example

Input Operations:
Insert (1, Alice, 85)
Insert (2, Bob, 91)
Update Bob's marks to 95
Delete Alice

Output:
Student inserted successfully
Record updated successfully
Record deleted successfully

Constraints

- Use PreparedStatement
- Handle SQL exceptions
- Follow best practices for closing connections

Bonus Challenge (+5 Marks)

Implement a Java feature from Java 11+ or 17 in any of the above problems. Example:
- var keyword
- Records
- Text Blocks
- Switch expressions
- Pattern matching

Submission Guidelines

- Create one folder named Module1_Assignment_<YourName>_RollNo.
- Submission Medium
  - GitHub
  - Soft Scanned copy of your assignment
- Note: Hard copy submission is Mandatory

Evaluation Criteria

Component	Marks
Correctness	30
Code Quality & Readability	15

Use of Concepts (IO, Lambda…)	20
Theory Section	25
Bonus Features (Optional)	5
Total	100