University of Ruhuna
Faculty of Technology
Department of Informtion and Communication Technology

Advanced Programming Practicum
(ICT3122)

Lab Sheet 9 26th Feb 2024

Object Serialisation in Java

Objective: The objective of this lab is to gain hands-on experience with serialization and deserialization in Java. By completing this lab, you will understand how to serialize Java objects, handle inheritance and aggregation in serialization, manage static data members, and use transient variables.

1. **Serialization Basics**
    - Create a Java class named **Car** with the following attributes:
        - **make** (String)
        - **model** (String)
        - **year** (int)
        - **price** (double)
    - Implement serialization for the **Car** class to serialize the object into a file named **car.ser**.
    - Write a method to deserialize the **Car** object from the **car.ser** file and display its details.
2. **Handling Inheritance in Serialization**
    - Create a superclass **Vehicle** with attributes **make** and **model**.
    - Create a subclass **Car** extending **Vehicle** with additional attributes **year** and **price**.
    - Implement serialization for the **Vehicle** and **Car** classes together, serialize a **Car** object, and deserialize it.
3. **Serialization with Aggregation**
    - Create a class **Engine** with attributes **serialNumber** (String) and **horsepower** (int).
    - Modify the **Car** class to contain an instance of **Engine**.
    - Implement serialization for the modified **Car** class with aggregation and test serialization and deserialization.

4. **Handling Static Data Members**
    - Add a static data member **totalCars** to the **Car** class to keep track of the total number of cars.
    - Serialize and deserialize **Car** objects while ensuring the static data member retains its value across serialization and deserialization.
5. **Using Transient Variables**
    - Add a transient variable **owner** (String) to the **Car** class.
    - Serialize and deserialize a **Car** object containing the transient variable and observe its behavior.

Instructions:

- Use proper exception handling techniques.
- Ensure the code is properly formatted and documented.
- Test each task separately and ensure the expected output is achieved.
- Feel free to refer the provided content and Java documentation for guidance.

Submission:

- Submit the Java source code files along with any additional files created during the lab.
- Include a brief report documenting the tasks completed, challenges faced, and any observations or insights gained during the lab.