Software Lab 1 Assignment 2

Debuggers

Scope	**Remarks**
Debuggers	Exploring the functionalities of GDB, DDD and Valgrind

• Problem statements

1.1 GDB

: Compile the **inventory.cpp** and complete the following tasks:

1. **Analyse the crash**:
   - Run the program with the update argument inside GDB.
   - When it crashes, use the backtrace command to find the exact function and line number causing the crash.
   - Inspect the values of local variables in that stack frame to understand why the crash occurred.

2. **Use a Conditional Breakpoint**:
   - The update_inventory function has special logic for the item with ID 103. Set a breakpoint in that function that only stops when it is processing that specific item.
   - When the breakpoint hits, print the item's quantity. Then step forward past the modification and print it again to verify the logic.

3. **Use a Watchpoint**:
   Set a watchpoint on **g_items_updated** to report how it increments. Continue execution and report the GDB's behaviour as observed.

4. **Explore the Call Stack**:
   - Run the program in GDB with the validate argument.
   - Set a breakpoint on the **generate_validation_code** function.
   - As you step through the function, use backtrace to report how the call stack grows with each recursive call.

1.2 Valgrind

Compile the **processor_challenge.cpp** and complete the following tasks:

1. **Find Memory Errors**:
   - Uninitialized Value.
   - Heap Overflow.
   - Memory Leak.
   - Invalid Free.

2. **Find Threading Errors**.

3. **Profile the Code**.

• Evaluation Date

- **Section A and B** : First week of September 2025

• Evaluation Criteria

Total Marks: 10 marks

- **Working examples**: 5 marks
- **Viva**: 5 marks