Objective:

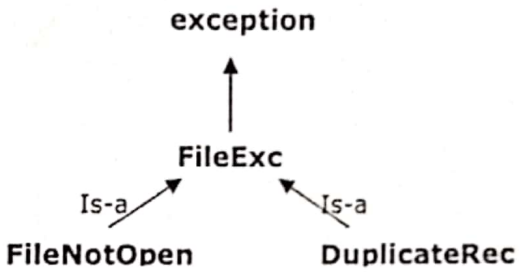
- Use of exception handling mechanism in our classes.

Task – 1:

Apply Exception Handling feature in GenericArray class discussed in the class as well. In addition to that, pick classes (that you implemented in OOP) of your choice to apply exception handling.
Hint: search out for exit(0) in your code.

Task – 2:

Pick any of your file handling code of OOP: the function, which is responsible for opening a file, should throw an exception of type *FileNotOpen* exception if file doesn't get opened. Similarly, the function, which is adding records in file and is responsible for maintaining uniqueness, throws an exception of type *DuplicateRec,* if duplicate record is found.
Implement the following exception hierarchy for this task.



Task – 0: ***C++ exception hierarchy***

All exceptions generated by the C++ standard library inherit from **`std::exception`**.
May explore their details from the link: https://en.cppreference.com/w/cpp/error/exception

- logic_error
  - invalid_argument
  - domain_error
  - length_error
  - out_of_range
  - future_error(C++11)
- bad_optional_access(C++17)
- runtime_error
  - range_error
  - overflow_error
  - underflow_error
  - regex_error(C++11)
  - system_error(C++11)
    - ios_base::failure(C++11)
    - filesystem::filesystem_error(C++17)
  - tx_exception(TM TS)
  - nonexistent_local_time(C++20)
  - ambiguous_local_time(C++20)
  - format_error(C++20)
- bad_typeid
- bad_cast
  - bad_any_cast(C++17)
- bad_weak_ptr(C++11)
- bad_function_call(C++11)
- bad_alloc
  - bad_array_new_length(C++11)
- bad_exception
- ios_base::failure(until C++11)
- bad_variant_access(C++17)