BOOK MANAGEMENT – CASE STUDY

1. Creating a JAVA Project

- Set up JAVA Development Environment by installing JAVA Development Kit (JDK) and an Integrated Development Environment (IDE) like Eclipse.
- Create a new JAVA Project
    - Inside the *./src* file create a new package *"com.java.casestudy".*
    - Create 4 *.java* file inside this package. These are:
        - *Main.java* – Main console-based application program.
        - *Book.java* – Class to represent individual books with attributes ISBN and Book Name and methods like constructor and getter – setter functions.
        - *Library.java* – Class to manage the collection of books.
        - *BookNotFoundException.java* – Implements error handling to deal with scenarios like incorrect user input.
    - Built-in library used are *java.util.** and *java.io.**

2. Implementing Book Management

- Created a menu that allows users to navigate through the available options easily.
- It is implemented through do – while loop and switch statements.
- Try – catch block is implements to handle the possible incorrect user inputs like validity of ISBN number and book details not found exceptions.
- Three overloaded functions are defined inside the '*Main.java'* class to use the Custom Exception class '*BookNotFoundException.java'*.
- File IO is handled in '*Main.java'* class through *'serialize(Library _library, String _fileName)'* and *'deserialize(String _fileName)'* functions.

3. Handling Data Storage and Retrieval

- The records of all the books are stored in a '*Library'* object having the name '*library'*.
- This object has the details of all the books stored inside a *HashSet* of type *Book* defined inside the '*Library.java'* class.
- The '*library'* object is stored (serialized) and retrieved (deserialized) in/from a *'records.ser'* file by implementing *java.io.Serializable*

4. Implementing CRUD Operations

- Create Book Objects and add Books to the Library:
    - ISBN and book name are stored in an object when object of type '*Book'* is created.
    - Implements a function '*registerBook(Book _bookObj)'* to add newly created '*Book'* object to the '*Library'* object.
- Read Book Details:
    - Implements a function '*retireveBookGivenIsbn(String _isbn)'* to find books by their ISBN.

    - Once the book is found display the ISBN and book name.
- Update Book Details:
    - Implements a function *'updateBookDetails(String _isbn, String _updatedBookName)'* to update the details of a specific book.
    - Internally it make use of '*retireveBookGivenIsbn(String _isbn)'* to find the existence of the specific book in the record.
- Delete Books:
    - Implements a function *'deleteBook(String _isbn)'* to remove a book from the '*library'* based on the ISBN.
    - Internally it make use of '*retireveBookGivenIsbn(String _isbn)'* to find the existence of the specific book in the record.