

|  |  |  |
| --- | --- | --- |
| GROUP MEMBERS | REGISTRATION NUMBER | STUDENT NUMBER |
| MUZAHURA JOSHUA | B/23/U/D0611/PS | 2300900611 |
| NAMWEMEZI PARASIDE | B/23/U/D0614/PS | 2300900614 |
| AGABA RICHARD | B/23/U/D0954/PS | 2300900954 |

**Class Definition: `Book`**

1. Purpose: This class represents a book with attributes for the title, author, and publication year.

2. Constructor (`\_\_init\_\_` method): Initializes a `Book` object with three parameters:

- `title`: The title of the book.

- `author`: The author of the book.

- `year\_published`: The year the book was published.

3. Method `describe`: Returns a formatted string that describes the book in the format:

"<title> by <author>, published in <year>"

**Function: `sort\_books\_by\_year`**

1.Purpose: Sorts a list of `Book` objects by their publication year.

2. Parameters:

- `books`: A list of `Book` objects to be sorted.

3. Returns: A sorted list of `Book` objects based on the `year\_published` attribute.

4. Implementation:

- If the `books` list is empty, it returns an empty list.

- Otherwise, it uses Python's `sorted()` function to sort the books by `year\_published`.

---

**Creating and Using Book Objects**

1. Instantiation: Five `Book` objects are created, each representing a classic book with details:

- `"The Lord of the Rings"` by J.R.R. Tolkien (1954)

- `"Pride and Prejudice"` by Jane Austen (1813)

- `"Oliver Twist"` by Charles Dickens (1863)

- `"Frankenstein"` by Mary Shelley (1818)

- `"Beloved"` by Toni Morrison (1987)

2. Describing Books: The `describe` method is called on each book to print a description.

---

**Sorting and Looping Through Books**

1. Sorting: The `books` list is passed to `sort\_books\_by\_year`, and the sorted list is stored in `sorted\_books`.

2. For Loop:

- Iterates through `sorted\_books` and prints the title, author, and year of each book.

3. While Loop:

- Accomplishes the same task as the for loop using a while loop with an index counter.

---

**Interactive Book Search**

1. Purpose: Allows the user to search for a book by title.

2. Implementation:

- Repeatedly prompts the user to enter a book title to search for.

- If the user types "exit," the loop breaks.

- The code searches `sorted\_books` for a match and prints the book's details if found, or a "Book not found" message if not.

---

**User Interaction: Name, Age, and Favorite Number**

1. Name Input:

- Asks the user for their name and greets them.

2. Age Input:

- Asks the user for their age and calculates their birth year, using error handling for non-numeric input.

3. Favorite Number Input:

- Asks the user for their favorite number.

- Determines if the number is even or odd.

- Further checks if the number is greater than or less than 10, or exactly 10, and provides feedback.

**Error Handling**

1. ValueError Handling:

- If a non-numeric input is given for age or favorite number, the code prints an error message.

This code combines object-oriented programming (with the `Book` class) and user interaction, showcasing sorting, looping structures, and input validation.