Abstraction

**Activity A: Library management system**

Create a program that manages a small library of books. Users should be able to:

1. Add new books.
2. Search for a book by title.
3. Remove a book.
4. Display all books.

Task 1: Decomposition

Using the grid provided, break the main problem into smaller, manageable parts. Each box represents a task and one has been done as an example.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Book Storage – How will books be stored? (e.g., a list or database) |  |  |  |  |

Task 2: Abstraction

What information about the books can be ignored and what information about the books is important and required to solve the problem?

|  |  |
| --- | --- |
| **Ignored/Removed** | **Important/Kept** |
|  |  |

Task 3: Solve each part systematically

The following code can be used to store the books.

library = [] # A simple list to store books

Write an algorithm using a subroutine for each one of the sub-tasks covered in Task 1.

|  |  |
| --- | --- |
| **Add book** |  |
| **Search book** |  |
| **Remove book** |  |
| **Display book** |  |

Task 4: Combine the solutions

Use the subroutines created to add **1984 by George Orwell** and display the results.

|  |
| --- |
|  |

**Activity B: Calculator program**

A program for a simple calculator that can perform addition, subtraction, multiplication, and division.

Task 1: Abstraction

Decompose the program into **four** modular parts.

Task 2: Structure program into modular parts

1. Write an algorithm that uses a subroutine called Division. It must take two values as parameters and either return the answer or returns an error if the second number is 0.

|  |
| --- |
|  |

1. Write an algorithm that will use a subroutine called Calculator which will:

* Take two numbers are parameters
* Allow the user to enter the arithmetic operator
* Perform the operation against the two numbers by calling the appropriate module (e.g. Addition, Subtraction etc..) and returning the result.

|  |
| --- |
|  |