# Requirements Analysis Worksheet

## 1. Project Selection

Chosen Project 1: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Chosen Project 2: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 2. Problem Definition

• What problem does this program solve?

• Who will use this program?

• Why is this solution needed?

## 3. User Requirements

• What features do users need?

• How should users interact with the program?

• Example: 'Users should be able to add and delete tasks in a To-Do List.'

## 4. Functional Requirements

List the core functionalities of your program. Examples:  
• The Temperature Converter should allow users to input a value and select conversion.  
• The Contact Book should store, retrieve, and update contacts.

• Feature 1: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

• Feature 2: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

• Feature 3: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 5. Non-Functional Requirements

Consider aspects like:  
• Performance (e.g., 'The program should process inputs in under 1 second.')  
• Security (e.g., 'Contact Book data should be encrypted.')  
• Usability (e.g., 'The GUI should be simple and intuitive.')

## 6. Inputs & Outputs

Identify what the user will input and what the system will output. Examples:  
• Temperature Converter: User inputs a temperature and unit; system outputs converted value.  
• Calculator: User inputs two numbers; system outputs the result.  
• Expected Inputs: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

• Expected Outputs: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 7. Constraints & Assumptions

• Are there any technical limitations? (e.g., 'Program runs only on Windows.')  
• Are there any assumptions about the user? (e.g., 'User understands basic arithmetic for the calculator.')