Operational Specification Template Instructions

Purpose	- To hold descriptions of the likely operational scenarios followed during program use - To ensure that all significant usage issues are considered during program design - To specify test scenarios
General	- Use this template for complete programs, subsystems, or systems. - Group multiple small scenarios on a single template, as long as they are clearly distinguished and have related objectives. - List the major scenarios and reference other exception, error, or special cases under comments. - Use this template to document the operational specifications during planning, design, test development, implementation, and test. - After implementation and testing, update the template to reflect the actual implemented product.
Header	- Enter your name and the date. - Enter the program name and number. - Enter the instructor's name and the programming language you are using.
Scenario Number	Where several scenarios are involved, reference numbers are needed.
User Objective	List the users' likely purpose for the scenario, for example, to log onto the system or to handle an error condition.
Scenario Objective	List the designer's purpose for the scenario, for example, to define common user errors or to detail a test scenario.
Source	- Enter the source of the scenario action. - Example sources could be user, program, and system.
Step	Provide sequence numbers for the scenario steps. These facilitate reviews and inspections.
Action	Describe the action taken, such as - Enter incorrect mode selection. - Provide error message.
Comments	List significant information relating to the action, such as - User enters an incorrect value. - An error is possible with this action.

Functional Specification Template Instructions

Purpose	- To hold a part's functional specifications - To describe classes, program modules, or entire programs
General	- Use this template for complete programs, subsystems, or systems. - Use this template to document the functional specifications during planning, design, test development, implementation, and test. - After implementation and testing, update the template to reflect the actual implemented product.
Header	- Enter your name and the date. - Enter the program name and number. - Enter the instructor's name and the programming language you are using.
Class Name	- Enter the part or class name and the classes from which it directly inherits. - List the class names starting with the most immediate. - Where practical, list the full inheritance hierarchy.
Attributes	- Provide the declaration and description for each global or externally visible variable or parameter with any constraints. - List pertinent relationships of this part with other parts together with the multiplicity and constraints.
Items	- Provide the declaration and description for each item. - Precisely describe the conditions that govern each item's return values. - Describe any initialization or other key item responsibilities.
Example Items	An item could be a class method, procedure, function, or database query, for example.

Logic Specification Template Instructions

Purpose	- To contain the pseudocode for a program, component, or system - To enable precise and complete program implementation - To facilitate thorough design and implementation reviews and inspections
General	- Use this template to document the program's detailed logic. - After implementation and testing, update the template to reflect the actual implemented product. - During detailed design, write the pseudocode needed to describe all of the program's logic. - Use plain language and avoid using programming instructions wherever practical.
Header	- Enter your name and the date. - Enter the program name and number. - Enter the instructor's name and the programming language you are using.
Design References	List the references used to produce the program's logical design. - the Operational, Functional, and State templates - the program's requirements - any other pertinent source
Parameters	- Where needed, define any parameters or abbreviations used. - Avoid duplicating definitions on other templates and reference these other definitions where they are needed.