Functional Specification Template

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Student** | | |  | | **Date** |  |
| **Program** | | |  | | **Program #** |  |
| **Instructor** | | |  | | **Language** |  |
|  | | | | | | |
| **Class Name** | |  | | | | |
| **Parent Class** | |  | | | | |
|  | |  | | | | |
|  | |  | | | | |
|  | |  | | | | |
|  | | | | | | |
| **Attributes** | | | | | | |
|  | **Declaration** | | | **Description** | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  | | | | | | |
| **Items** | | | | | | |
|  | **Declaration** | | | **Description** | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |

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. |