# Project Component

*A project is separated into individual components. You will use this document to design an individual component for a program.*

## Component Requirements

### Problem to be Solved

*Why is this component needed?*

### Constraints

*What are the boundaries and limitations of this project?*

### Benefits

*What advantage is there in developing this component?*

*What advantage would it bring to a project?*

### Nature of Interactivity

*What functionality will it provide for a project?*

*How will the program or users interact with it?*

### Complexity of Problems

*How complex is the problem you are trying to solve?*

*For example: Is there some advanced mathematics, or a large number of classes that need developing.*

## Test Planning

*The test plan is completed up to expected results only during the design phase.*

*During the development phase the actual results are added, and the comments are completed.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Number** | **Purpose of test** | **Test Data** | **Expected Result** | **Actual Result** | **Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Structure Diagram

*The general structure for the component. This should include interfaces, parent and child classes, as well as any controlling class. Describe the structure and how it links to the component requirements.*

### Module Descriptions

#### Interface A

Interface A has the purpose of…

#### Interface B

Interface B has the purpose of…

#### Parent Class A

Parent Class A has the purpose of…

Abstract Class A

Implements Interface A

Child Class C

Inherits Class A

Implements B

Child Class D

Inherits Class A

Interface A

Interface B

Controller A

Parent Class B

## Class table: Abstract Class A

### Class Description

### Fields

**Private** *var* **As** *DataType*

### Properties

**Public Read Write** pVar() **As** *DataType*

### Constructor

**Public New** (*paras* **As** *DataType*)

### Methods

funcA(*paras* **As** *DataType*) **As** *DataType*

subA(*paras* **As** *DataType*)

### Pseudocode

‘Constructor for the class.

‘Preconditions: The information it needs to work.

‘Postconditions: None.

**Public Constructor New** (*paras* **As** *DataType*)

‘Code goes here

**End Constructor**

‘Preconditions: The information it needs to work.

‘Postconditions: The value it returns.

**Public Function** funcA(*paras* **As** *DataType*) **As** *DataType*

‘Code goes here

**End Function**

‘Preconditions: The information it needs to work.

‘Postconditions: None.

**Public Sub** subA(*paras* **As** *DataType*)

‘Code goes here

**End Sub**

## Class table: Child Class B **Inherits** Parent Class A

### Class Description

### Fields

**Private** *var* **As** *DataType*

### Properties

**Public Read Write** pVar() **As** *DataType*

### Constructor

**Public New** (*paras* **As** *DataType, paras* **As** *DataType*)

### Methods

**Overloads** funcA(*paras* **As** *DataTypes, paras* **As** *DataTypes*) **As** *DataType*

subB(*paras* **As** *DataType*)

### Pseudocode

‘Constructor for the class.

‘Preconditions: The information it needs to work.

‘Postconditions: None.

**Public Constructor New** (*paras* **As** *DataType*)

‘Code goes here

**End Constructor**

‘Preconditions: The information it needs to work.

‘Postconditions: The value it returns.

**Public Overloads Function** funcA(*paras* **As** *DataType*) **As** *DataType*

‘Code goes here

**End Function**

‘Preconditions: The information it needs to work.

‘Postconditions: The value it returns.

**Public Sub** subB(*paras* **As** *DataType*)

‘Code goes here

**End Sub**

## Class table: Child Class C **Inherits** Parent Class A

### Class Description

### Fields

**Private** *var* **As** *DataType*

### Properties

**Public Read Write** pVar() **As** *DataType*

### Constructor

**Public New** (*paras* **As** *DataType, paras* **As** *DataType*)

### Methods

funcB(*paras* **As** *DataTypes, paras* **As** *DataTypes*) **As** *DataType*

subC()

### Pseudocode

‘Constructor for the class.

‘Preconditions: The information it needs to work.

‘Postconditions: None.

**Public Constructor New** (*paras* **As** *DataType*)

‘Code goes here

**End Constructor**

‘Preconditions: The information it needs to work.

‘Postconditions: The value it returns.

**Public Overloads Function** funcA(*paras* **As** *DataType*) **As** *DataType*

‘Code goes here

**End Function**

‘Preconditions: The information it needs to work.

‘Postconditions: The value it returns.

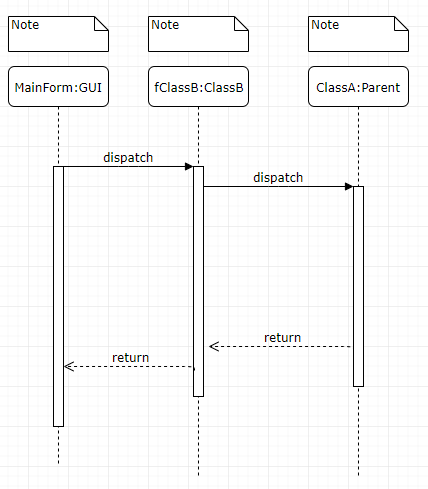
**Public Function** subC()

‘Code goes here

**End Function**

## Data Flow Diagrams

These show the passing of messages between different classes.



## Flow Charts

*These show processes inside each subroutine.*