No coding should be before Test Plan Version 2 is complete.

Part 1 is steps up to and including Test Plan Version 2.

Part 2 turn-in INCLUDES Part 1 plus additional information after Test Plan Version 2.

1. **Problem Statement**

Problem Statement is used to state the specific problem that needs to be solved by the program being written

1. **Requirements**
   1. **Assumptions**

What is assumed about the problem above that does not need consideration

* 1. **Specifications**

What is specifically required to solve the problem above

1. **Decomposition Diagram** (Used to break program down into components visually. Can have as many components as needed. Defines functionality that will solve the problem – does NOT define a flow )

Main

Component #2

Component etc.

Component etc.

Component #1

Component etc.

Component #1

Input

Output

Process

Component #1

1. **Test Strategy**

What type of testing is going to be done?

examples: Invalid data, Valid data, or File Testing (exist, empty, etc.)

1. **Test Plan Version 1**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Strategy | Test Number | Description | Input | Expected Output | Actual Output | Pass/Fail |
| What type of test is this specific case? | What number is this specific case? | What is the purpose of this specific case? |  |  |  |  | |
|  | Can have as many test cases that you can think of. | Do not fill in any more test cases once you start writing the Initial Algorithm  As they are put in version 2  Do not fill in columns past this point |  |  |  |  | |

1. **Initial Algorithm**

Your first algorithm should go here. DO NOT CHANGE ANYTHING ONCE YOU START CODING!!! What is the logic to follow in the program to solve the problem?

1. **Test Plan Version 2**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Strategy | Test Number | Description | Input | Expected Output | Actual Output | Pass/Fail |
| Copy and Paste from Version 1 |  |  | What is the input of this case? | What do you expect the output to be? |  |  |
|  | Add any test cases you thought of when making your Initial Algorithm and HIGHLIGHT THEM |  |  | Do not fill in columns past this point |  |  |

Part 1 ends here!!!!!!

1. **Code**

Copy and paste your code here. MAKE SURE TO COMMENT YOUR CODE!

A baseline for commenting is before any function add this:

//Description: What does the function do

//Pre-condition: What do input do you need for the function to work

//Post-condition: What is the end result of the function or what do you get out of the function

Also the beginning of your program should have these comments:

//Program Name:

//Programmer Name:

//Description:

//Date Created:

1. **Updated Algorithm**

Copy and paste Initial Algorithm and make any updates to reflect the changes you made in your code. HIGHLIGHT THE CHANGES YOU MAKE! Strike out deleted statements. Any statements that just have a wording change – make change and highlight (i.e. no need to strike out individual word changes). This is the FINAL documentation of your program and needs to match what code you created.

1. **Test Plan Version 3**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Strategy | Test Number | Description | Input | Expected Output | Actual Output | Pass/Fail |
| Copy and Paste from Version 2 | Un-highlight any test cases are from version 2 that were highlighted |  |  |  | What was the actual output from your code | If your actual output matches with expected output, write PASS otherwise write FAIL |
|  | Add any final test cases you could think of and HIGHLIGHT THEM |  |  |  |  | Any test cases that fail you must change your code to make the cases pass |

1. **Screenshots**

Screenshots of your testing goes here. YOU MUST HAVE A SCREENSHOT FOR EVERY TEST CASE. A screenshot may picture multiple test cases. For each screen shot caption it with a list of the test cases are depicted in it.

1. **Error Log**

Any issues you had while testing your code are recorded in the error log as you perform testing of the “completed” code – that is, when you run through all of the test cases in the test plan.

|  |  |  |
| --- | --- | --- |
| Error Type | Cause of Error | Solution to Error |
| Log 2 types of errors:  Logic  Runtime | What specifically caused the error to occur | What did you do/change to fix the error |
|  |  |  |

Do not list any syntax errors or errors detected in unit testing as you build your program.

1. **Status**

What is the final status of your program? Does it fully work? Are there any test cases that fail and if so which ones? What needs to be done to correct the defects?