Program Assignment Requirements

Your program must meet the following criteria:

* Have an UML Diagram that adheres to the UML syntax
* Your program must compile; programs that do not compile will not be graded and a zero will be applied to the grade
* At the beginning of your program (or the “test” program), you have to have the following:
* **State your name**
* State the Java Version that is being used
  + Java #
* State what the program does
  + This program creates …..
* You must list the name of all the files that are necessary to have the run correctly (list all the files used in the program solution) and then state which file is the file to execute to run the program
  + This program needs file1.java, file2.java; execute file1.java to run the program
* State where the input is from to run the program
  + From the keyboard, hard coded, from the file name, …
* State where the output is displayed
  + To the screen, file name, …
* Provide a sample run of the program
* Your program must adhere to the Java standards and conventions:
  + Variables used in your program should follow Java’s naming conventions
  + Your program should have proper indentations, spacing of blocks of code, and block styles (NO spaghetti code!)
  + Your program should have appropriate comments when/where applicable
* Your program should solve the stated objectives of that program
* Your program should produce a valid output (invalid outputs are a worthless program = grade 0)
* Your program should use the current techniques and concepts that we are covering

Program Grading Rubric

|  |  |
| --- | --- |
| **Requirement** | **Points** |
| UML | 10 |
| Compiles = graded  Does not compile = 0 |  |
| Required Intro Comments | 5 |
| Programming Style & Documentation | 2 or 3 |
| Program solves stated objective(s) | 5 |
| Program produces output that is correct / valid | 5 |
| Program uses current topics ideas / techniques | 5 |
| Proper file names |  |
| Miscellaneous | 2 or 3 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | UML | Required Intro Comments | Programming Style & Documentation | Program solves stated objective(s) | Program produces output that is correct / valid | Program uses current topics ideas / techniques | Miscellaneous | Proper file names |
| Concept is filled in its entirety | 10 | 5 | 3 | 5 | 5 | 5 | 2 |  |
| Missing 1 – 2 aspects of the concept | 8 | 4 | 2 | 3 - 4 | 3 - 4 | 3 - 4 | 1 | -1 |
| Missing 3 – 4 aspects of the concept | 5 | 2 | 1 | 1 - 2 | 1 - 2 | 1 - 2 | 0 | -2 |
| Missing too many aspects of or does not have the concept | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -5 |

Example: