Southern New Hampshire University

CS 320 Milestone One Guidelines and Rubric

Overview: The final project for this course is a summary and reflections report documenting the decision making and results of your tests for the Medical Application. You will submit the software tests that you develop in this milestone, which will inform your final project. Refer to the Final Project Code zip file and the Final Project Test Plan document to complete this assignment.

In Module Three, you completed a tutorial that demonstrated two possible approaches to software testing. For your milestone, you can use either of these approaches, but be sure to consider the best practices you have learned as well as the requirements in the test plan.

Prompt: Your code should meet each of the following criteria:

Technically Sound

- Your code should be **syntactically accurate**. This is a measure of the style and syntax of your program code. This will include factors such as the following:
  - Statement choice (e.g., for vs. while, nested vs cascaded if-else, appropriate use of Boolean operators, fields vs. variables)
  - Appropriate use of techniques to enhance reusability and maintainability of code (e.g., class constants, private helper methods, proper access modifiers)
- Your code should be **logical.** This is a measure of how well your program satisfies the requirements of the assignment. This may include the following:
  - Checking that your code meets the specifications in the assignment (e.g., specific class names and method signatures)
  - Running your code against the JUnit reference tests
  - Executing your program manually and checking that it performs all required operations correctly (if not fully tested by the JUnit reference tests)
  - Inspecting your source code for bugs not uncovered by other testing methods
  - Ensuring proper methods are used, including appropriate tags (e.g., @param, @return, @before)

Efficient

- Your code should be **concise.** This is a measure of how thoroughly you have tested your own code. This will include the following factors:
  - Code does not perform unnecessary work (no extraneous fields/variables/code, data type selection, etc.)
  - Code is tested thoroughly (all code is exercised, likely sources of errors are probed, boundary cases are tested, etc.)
- Your code should be **modular.** This is a measure of how you organized your tests (e.g., grouping tests into test suites) for each of the features. This will include the following:
  - Tests are organized to maximize testing for each of the features identified in the requirement document.
  - The purpose of each test is readily identifiable based upon the test method name and/or added comments.
  - Each test includes appropriate assertions with helpful failure messages.

Additionally, your tests **overall** should be **effective**. This means that they must accurately identify **at least five bugs** in the application and indicate the nature of each of the identified bugs.

Rubric

Guidelines for Submission: You will submit your code as a zip file containing your tests.

Critical Elements	Proficient (100%)	Needs Improvement (70%)	Not Evident (0%)	Value
Syntactically Accurate Code	Code is syntactically accurate	There are minor issues of syntactical accuracy, but they do not fully compromise the tests	There are significant issues of syntactical accuracy that fully compromise the tests	15
Logical Code	Code is functionally logical	There are minor issues of functional logic, but they do not fully compromise the tests	There are significant issues of functional logic that fully compromise the tests	15
Concise Code	Code is appropriately concise, yielding efficient tests	There are minor issues where the code is not appropriately concise resulting in some loss of efficiency	The conciseness of the code is a systemic issue that significantly impacts the efficiency of the tests	15
Modular Code	Code is appropriately modular, yielding efficient tests	There are minor issues where the code is not appropriately modular resulting in some loss of efficiency	The modularity of the code is a systemic issue that significantly impacts the efficiency of the tests	15
Overall Effectiveness of the Tests	Tests accurately identify at least five bugs	Tests accurately identify fewer than five bugs, or not all identified bugs are accurate	Tests do not accurately identify any bugs	40
			Total	**100%**