

|                            | White Box  |  |                                     |                          |
|----------------------------|--|--|-------------------------------------|--------------------------|
| Defect Number              | Defect Description   | How to Correct the Defect  | Module in Which the Defect is Found | Severity (S1, S2, or S3) |
| InstructorFirstScreen.java |  |  |                                     |                          |
| 1                          | The code in the action listener of "Grade Test" button calls the function to create a new test.  | Edit code in the action listener of the "Grade Test" button to grade tests.  | Initial Instructor GUI              | S1                       |
| 2                          | The code in the action listener of "EditTest" button calls the function to grade tests.  | Edit code in the action listener of the "Edit Test" button to edit tests.  | Initial Instructor GUI              | S1                       |
| 3                          | The code in the action listener of "Create Test" button calls the function to edit tests.  | Edit code in the action listener of the "Create Test" button to create tests.  | Initial Instructor GUI              | S1                       |
| 4                          | The author used <code> javax.swing.JButton </code> instead of <code> JButton </code> . This makes the code looks very messy.   | Add <code> import javax.swing.*; </code> at the beginning of the code.   | Initial Instructor GUI              | S3                       |
|                            |  |  |                                     |                          |
| InstructorTestify.java     |  |  |                                     |                          |
| 5                          | The function <code> fileChooserActionPerformed </code> is empty -- after the user chooses a file, there is no action done to read the file or handle errors.   | Add code to read files and handle errors.  | Instructor Create/Edit Test GUI     | S1                       |
| 6                          | In the function <code> nextQButtonActionPerformed </code> , the index for the current questions is decreased.  | Since the function is meant to load the next question, the index for the current question should be incremented.                         | Instructor Create/Edit Test GUI     | S1                       |
| 7                          | In the function <code> saveQuestion </code> , it did not handle errors correctly. For errors like "no total points possible is set for the question", it only displays an error dialog. It did not prompt for the correct total points possible. | Prompt to get the information if it is not already set.  | Instructor Create/Edit Test GUI     | S1                       |
| 8                          | In the function <code> saveQuestion </code> , it did not save if the question type is short answer.  | Add code to save question if the question type is short answer.  | Instructor Create/Edit Test GUI     | S1                       |
| 9                          | In the function <code> showProperties </code> , the program shows the same property dialog twice.  | The program should only show the property dialog once.   | Instructor Create/Edit Test GUI     | S1                       |
| 10                         | In the function <code> openExam </code> , it created a new test if the file of an exam cannot be loaded correctly.   | The program should prompt for another file unless the instructor presses "cancel" button.  | Instructor Create/Edit Test GUI     | S2                       |
| 11                         | In the function <code> newSAActionPerformed </code> , the program stops adding new question if the number of questions reaches 50.   | The maximum number of questions allowed in a test should be able to be set by the instructor instead of being hard coded in the program. | Instructor Create/Edit Test GUI     | S2                       |

|                     |   |  |                                 |    |
|---------------------|---|--|---------------------------------|----|
|                     | 12 In the function  saveQuestion , it only checked the error condition when the current index is less than 0; it should also check for the condition when the index is greater than or equal to the length. | if (current < 0    current > test.length()) {return;}                          | Instructor Create/Edit Test GUI | S2 |
|                     | 13 In the function  newTest , total points possible is set to 5 as default.   | Total points possible should be set to 0 as default.                           | Instructor Create/Edit Test GUI | S3 |
|                     |   |  |                                 |    |
| StudentTestify.java |   |  |                                 |    |
|                     | 1 Lines 192 -210 contain a hard coded question that will be present in every exam, the point value is also used for every question of every exam  | Point value should be changed to be dynamic, everything else should be removed | Student Client GUI              | S1 |
|                     | 2 Lines 246-300 contain a hard coded multiple choice question   | This should be removed   | Student Client GUI              | S2 |
|                     | 3 lines 354 - 380 contain a hard coded programming question   | this should be removed   | Student Client GUI              | S2 |
|                     | 4 on line 530 there is no code for what to do if the type of question is a multiple choice, there is no error message, but the answer will not be saved   | add a similar statement as the above to if statements to set the answer        | Student Client GUI              | S1 |
|                     | 5 On line 649, the points earned are set to be one less than they should  | remove the '-1'  | Student Client GUI              | S1 |
|                     | 6 On line 674, Only compiles every other file.  | remove the if statement, it is only there to cause a bug                       | Student Client GUI              | S1 |
|                     | 7 "please enter Both a first and a last name" is the only error message, and is used for everything   | Error message should be changed to reflect the error that is occurring.        | Student Client GUI              | S3 |
|                     |   |  |                                 |    |
| GraderTestify.java  |   |  |                                 |    |

|                 |   |  |  |    |
|-----------------|---|--|--|----|
| 1               | "testFile" variable is not explicitly initialized in "initComponents()".  | added testFile to initComponents()   | Instructor Client GUI - Grade Test             | S3 |
| 2               | All of the menu items have a setMnemonic except autograde and OpenKeyMenuItem   | Added a .setMnemonic("g") and setMnemonic("k")   | Instructor Client GUI - Grade Test             | S2 |
| 3               | Extra functions that are empty such as "nextQButtonMouseClicked" and "exitMenuItemActionPerformed"  | For exitMenuItemActionPerformed added a System.exit(0); and got rid of nextQButtonMouseClicked, as well as when someone calls it | Instructor Client GUI - Grade Test             | S2 |
| 4               | *POSSIBLE DUPLICATE* functions are calling empty functions. mouseClicked calling prevQButtonMouseClicked which is an empty function.  | Got rid of the unnecessary functions and calls   | Instructor Client GUI - Grade Test             | S2 |
| 5               | questionTypeLabel is being changed unnecessary amount of times. It is first initialized to null and then to "jlabel8". The jLabel.setText is not needed                                 | removed the unnecessary setText  | Instructor Client GUI - Grade Test             | S3 |
| 6               | Not needed if statement " if (test.length() != key.length())" a little bit above that, that if statement is being tested and a proper error shows up                                    | removed it   | Instructor Client GUI - Grade Test             | S2 |
| 7               | There is a boolean called isNew. It keeps getting changed but is never used   | removed it   | Instructor Client GUI - Grade Test             | S2 |
| 8               | In saveAsMenuItemActionPerformed there is a if statement "if (testFile == null)" saveMenuItemActionPerformed(evt); This doesn't make sense as you should save if it's not an empty file | changed it to if(testFile !=null)<br>removed the else statement as well  | Instructor Client GUI - Grade Test             | S1 |
|                 |   |  |  |    |
| PropDialog.java |   |  |  |    |
| 1               | In function "compCheckboxActionPerformed()", there is no any implementation. As a result, the programming questions cannot be compiled.   | Implement the relevant code in the listener function.  | Instructor Client GUI - Create Test, Edit Test | S1 |
| 2               | In function "invalidateCheckboxActionPerformed()", there is no any implementation. As a result, the exam will not be tested for invalidation if the window loses focus.                 | Implement the relevant code in the listener function.  | Instructor Client GUI - Create Test, Edit Test | S1 |

|                    |   |   |  |    |
|--------------------|---|---|--|----|
|                    | 3 The variable "test" is local. In "saveProperties()" function, current properties are saved into the variable's attributes. As the variable is local (not global), old properties are overwritten with new properties everytime the constructor of "PropDialog.java" is called. In other words, the properties are not successfully saved. | Set the variable as global.   | Instructor Client GUI - Create Test, Edit Test   | S1 |
|                    | 4 Variables "timeFmt", "dateText" and "stText" are declared and initialized, but never used.  | Remove the variables.   | Instructor Client GUI - Create Test, Edit Test   | S3 |
|                    |   |   |  |    |
| SummaryDialog.java |   |   |  |    |
|                    | 1 Variable "totalEarned" is set to variable "earned". The variable "totalEarned" should be added with the variable "earned", not be set with.   | Change line 42 to "totalEarned += earned;"  | Instructor Client GUI - Grade Test               | S1 |
|                    | 2 Text value of "pctLabel" is depended on both variables "totalEarned" and "earned". As the variable "totalEarned" is wrong, the text value of "pctLabel" will be wrong as well.  | Verify the involved variables are calculated correctly to stop the butterfly effect.                          | Instructor Client GUI - Grade Test               | S1 |
|                    | 3 In "setModel()" function of JTable "table", the first argument ("new Object [[]]") is not initialized.  | Initialize the double array object with empty string or null at line 73.                                      | Instructor Client GUI - Grade Test               | S1 |
|                    | 4 Initialization of "types" and "canEdit" is repetitive. If more elements are needed (like hundreds of elements), repetitive initialization will be tedious.  | Use loop for initialization.  | Instructor Client GUI - Grade Test               | S3 |
|                    | 5 First argument of "isCellEditable()" function, "rowIndex" is useless.   | Remove the argument.  | Instructor Client GUI - Grade Test               | S3 |
|                    |   |   |  |    |
| Question.java      |   |   |  |    |
|                    | 1 The variables are not initialized.  | Initialize all string variables to "", and all integer variable to 0 in constructor to avoid exception.       | Instructor Client GUI (All) / Student Client GUI | S3 |
|                    | 2 String of format "h:mm a", if the variable hour is 10~12, there will be an exception.   | change the string format "h:mm a" to "hh:mm a" and modify corresponding code of Time(String str) constructor. | Instructor Client GUI (All) / Student Client GUI | S1 |
|                    | 3 Time didn't check the validation of data.   | Remind user to input integers instead of other type if the user input incorrectly.                            | Instructor Client GUI (All) / Student Client GUI | S1 |

|                     |   |  |  |  |                          |
|---------------------|---|--|--|--|--------------------------|
|                     | 1 | The number of compiles incremented when call getNumOfCompiles function.  | " return numOfCompiles; "instead of "return ++numOfCompiles;"  | Instructor Client GUI (All) / Student Client GUI | S1                       |
|                     | 2 | Comments are duplicated returned, in function getComments  | "return comment;" instead of "return comment +" "+" comment";  | Instructor Client GUI (All) / Student Client GUI | S2                       |
|                     |   |  |  |  |                          |
| MultipleChoice.java |   |  |  |  |                          |
|                     | 1 | The content of string array choices didn't pass to this choices.   | Assign choices to this.choices in function MultipleChoice  | Instructor Client GUI (All) / Student Client GUI | S1                       |
|                     |   |  |  |  |                          |
| Compile.java        |   |  |  |  |                          |
|                     | 1 | The argument of the function getProcess has not be used in the function.   | Correct the code of get property to fileName instead of os.name.   | Student Client GUI                               | S2                       |
|                     |   |  |  |  |                          |
| TestProperties.java |   |  |  |  |                          |
|                     | 1 | The class's functions will be carried out by its counterpart "PropDialog.java". In other words, the class does nothing and can be deleted without creating any error.              | Remove this class.   | N/A  | S2                       |
|                     | 2 | "initComponents()" function does not set format for date and time input fields. As a result, the user can insert arbitrary values into the fields.                                 | (If this class does perform something useful and cannot be removed) Implement code to set format for the date and time input fields. | N/A  | S1                       |
|                     |   |  |  |  |                          |
|                     |   | Black Box  |  |  |                          |
| Defect Number       |   | Defect Description   | How to Correct the Defect  | Module in Which the Defect is Found              | Severity (S1, S2, or S3) |
|                     | 1 | If a user clicks on "Create Test", a file chooser will pop up. If the user clicks on the "open" button in the file chooser, the program will quit unexpectedly.                    | Use  InstructorTestify.main(new String[] {""});  instead of  InstructorTestify.main(new String [] {"open"}); .                       | Instructor Client GUI - Create Test              | S1                       |
|                     | 2 | If a user clicks on "Create Test", a file chooser will pop up. If the user clicks on the "cancel" button in the file chooser, the program opens the menu for the edit test option. | The program should call  System.exit(0);  if the user clicks on the cancel button in the  openExam  function.                        | Instructor Client GUI - Create Test              | S1                       |

|    |  |  |                                     |    |
|----|--|--|-------------------------------------|----|
| 3  | Nothing happens when test properties option is selected.   | Add action listener for the test properties menu item.   | Instructor Client GUI - Create Test | S1 |
| 4  | You are able to modify fields without first opening or creating a new test.  | Prompt for a location to save the file at the beginning of the function  newTest .   | Instructor Client GUI - Create Test | S2 |
| 5  | If you try to move to the next question leaving the points possible field blank, you receive a warning, but can still move on (assigning severity 2 as the points can be "manually calculated and determined" afterwards). | The program only logs an error if the field is left as blank. It should also prompt for nonempty value in  saveQuestion  function.     | Instructor Client GUI - Create Test | S2 |
| 6  | Once the point field is modified, it stays the same for each question  | The pointsPossible variable should be set in the  loadQuestion  function.  | Instructor Client GUI - Create Test | S1 |
| 7  | When a new question of type multiple choice is selected, when you attempt to add choices, it adds the same choice twice  | The  addChoice  function is called twice. The program should call the function only once.  | Instructor Client GUI - Create Test | S1 |
| 8  | When adding choices for multiple choice, it allows you to add multiple copies of the same answer(both are entered twice per attempt, total of 4 identical answers after 2 identical add's)                                 | The actual choice for the multiple choice question is not checked. The program should check for duplicates in the choices.             | Instructor Client GUI - Create Test | S1 |
| 9  | When adding a multiple choice question, you are able to move on to next question without adding prompt, you get a warning, but it does allow you to continue   | The program only logs an error if the prompt is left as blank. It should also ask for nonempty value in  saveQuestion  function.       | Instructor Client GUI - Create Test | S1 |
| 10 | When adding a multiple choice question, If you attempt to add choices before entering a prompt, you get a warning, but the choices are still added (not sure if intentional)   | The program only logs an error if the prompt is left as blank. It should also ask for nonempty value in  saveQuestion  function.       | Instructor Client GUI - Create Test | S1 |
| 11 | When adding a multiple choice question, You are able to move on to the next question without completing the answer for the current multiple choice question  | The program only logs an error if the answer field is left as blank. It should also ask for nonempty value in  saveQuestion  function. | Instructor Client GUI - Create Test | S1 |
| 12 | When adding a multiple choice question, You are able to move on to the next question without completing choices for a multiple choice question   | The program only logs an error. It should also ask for complete set of nonempty values in  saveQuestion  function.                     | Instructor Client GUI - Create Test | S1 |
| 13 | When adding a multiple choice question, if you add an answer, then add a choice, the answer will disappear from the GUI  | The answer is not saved in the  saveQuestion  function. The program should save it as well.  | Instructor Client GUI - Create Test | S1 |
| 14 | When adding a short answer question, you are able to move on without adding a prompt, however a warning is displayed   | The program only logs an error. It should also ask for nonempty prompt in  saveQuestion  function.                                     | Instructor Client GUI - Create Test | S1 |

|    |   |   |                                     |    |
|----|---|---|-------------------------------------|----|
| 15 | When adding a short answer question, you are able to move on without adding an answer   | The answer is not checked in the  saveQuestion  function. The program should check whether it is empty as well.   | Instructor Client GUI - Create Test | S1 |
| 16 | When adding a programming question, you are able to move on without entering a prompt or a file name, however a warning is displayed  | The program only logs an error. It should also ask for nonempty prompt in  saveQuestion  function.  | Instructor Client GUI - Create Test | S1 |
| 17 | When adding a programming question, you are able to move on without adding an answer, no warning is displayed (assigning severity 2 as the instructor can choose to formulate the answers after completion of the test)         | The answer is not checked in the  saveQuestion  function. The program should check whether it is empty as well.   | Instructor Client GUI - Create Test | S2 |
| 18 | When adding a programming question, if you add a filename, then click next, the field remains filled out for the next question, but is uneditable if the next question is not a programming question                            | In the  saveQuestion  function, file name is not updated. It should be updated.   | Instructor Client GUI - Create Test | S1 |
| 19 | If you create a short answer, multiple choice or programming question, then move on, and return to the question, the answer has been removed from the GUI   | The answer is not checked or saved in the  saveQuestion  function. The program should save it.  | Instructor Client GUI - Create Test | S1 |
| 21 | If you create a short answer question, then return to the question using prev/next, or select question number, the prompt sometimes will be removed from GUI. The result seems inconsistent, and cannot replicate consistently. | If "previous" is clicked, The prompt loaded is the prompt of the next question; if "next" is clicked, The prompt loaded is the prompt of the previous question. This can be corrected by switching  current++  and  current-- . | Instructor Client GUI - Create Test | S1 |
| 22 | Next and previous buttons are switched, they take you in the opposite direction   | In function nextQButtonActionPerformed, current++; In function prevQButtonActionPerformed, current--;   | Instructor Client GUI - Create Test | S1 |
| 23 | If you cycle back through questions, the program warns you if you try to delete and leave a relevant field blank, but it still allows you to continue with a blank field  | The program only logs errors. In the  error  function, the program should handle error promptly based on the error code.  | Instructor Client GUI - Create Test | S1 |
| 24 | The program allows you to delete questions, but the drop down menu does not properly display the current question numbers   | The length of the  listModel  is decremented in the function. The function should return the length of the  listModel .   | Instructor Client GUI - Create Test | S1 |

|    |   |  |                                     |    |
|----|---|--|-------------------------------------|----|
| 25 | After questions are deleted, the drop down menu no longer works properly for the selecting a question to modify                   | The dropdown menu uses another list instead of  listModel . It should uses  listModel  as well.  | Instructor Client GUI - Create Test | S1 |
| 26 | When adding a point value to a question, the field stays the same when adding a new question, or switching to a previous question | The pointsPossible variable should be set in the  loadQuestion  function.  | Instructor Client GUI - Create Test | S1 |
| 27 | If you attempt to save a test with the same name as another test, the old test is automatically overwritten, with no warning      | Create a global list to store names of all saved tests. In "InstructorTestify.java" "saveQuestion()" function, before executing "question.setFile()", compare "fileTextField.getText()" with all names stored in the global list. If "fileTextField.getText()" records the same name as one of the names in the global list, pop up a warning message indicating the issue, and let the user to choose whether to overwrite the old test or not.   | Instructor Client GUI - Create Test | S1 |
| 28 | If you try to click "save as" twice in the same session, the "save as" menu does not appear                                       | The action listener for the "Save As" menu item does not prompt for the location of the file to save. Need to add file chooser in the action listener if file name is not specified.   | Instructor Client GUI - Create Test | S2 |
| 29 | If you try to save a file with no name, nothing happens, no warning is displayed  | In "InstructorTestify.java" "saveQuestion()" function, implement the code to test whether a file name is empty or not for all three types of questions (only programming-type questions' part is implemented). It can be done by implement the code repetitively for each of the question type, or as a general case applicable to all the question types. Note that nothing happens when the multiple-choice and short answer questions are saved due to internal logic of the program. | Instructor Client GUI - Create Test | S3 |



|    |  |  |                                     |    |
|----|--|--|-------------------------------------|----|
| 30 | If you try to open a previously saved test, the program opens a different test, but the contents are not correctly displayed | The program only updates the content when it is selected. It needs to reload all contents once a new test is opened.   | Instructor Client GUI - Create Test | S1 |
| 31 | If you select new test before saving the current test, your current progress is lost with no warning                         | The program does not save progress when new test is selected. In the action listener, it should check whether a test is being displayed.   | Instructor Client GUI - Create Test | S1 |
| 32 | Clicking on "Grade Test" button will open up a screen to grade tests.  | Correction: Clicking on "Grade Test" button will open up a screen to "Edit Test". In "InstructorFirstScreen.java" "gradeTestButtonActionPerformed(" function, call "GraderTestify.main()" instead of "InstructorTestify.main()". | Instructor Client GUI - Grade Test  | S1 |
| 33 | Although there is an option -- "Open Key" in the drop down menu, there is no way to create a key file for an exam.           | Correction: This is not a bug as the instructor needs to open an exam to load the key answers and then choose "Open Student Exam" to grade the exam.   | Instructor Client GUI - Grade Test  | S1 |
| 34 | After clicking on "Open Key", a file chooser will show up, clicking on a file yields no responses.                           | Correction: This is not a bug as the instructor needs to open an exam to load the key answers and then choose "Open Student Exam" to grade the exam.   | Instructor Client GUI - Grade Test  | S1 |
| 35 | After clicking on "Open Student Exam" and choosing an exam file, the program could not load exam information.                | Correction: This is not a bug as the instructor needs to open an exam to load the key answers and then choose "Open Student Exam" to grade the exam.   | Instructor Client GUI - Grade Test  | S1 |
| 36 | After clicking on "Open Student Exam" and choosing an exam file twice, the program could load exam information.              | Correction: This is not a bug as the instructor needs to open an exam to load the key answers and then choose "Open Student Exam" to grade the exam.   | Instructor Client GUI - Grade Test  | S1 |

|    |   |  |                                    |    |
|----|---|--|------------------------------------|----|
| 37 | The program does not have any responses after clicking on "Auto-grade multiple choice" button in the drop down menu, which should auto-grade multiple choice questions. | The "test" variable in "GraderTestify.java" is not initialized yet when "autogradeMenuItemActionPerformed" is called at line 217. It is initialized via function call at line 234. Change the order of the function calls to ensure the variable has been initialized before "autogradeMenuItemActionPerformed" is called.   | Instructor Client GUI - Grade Test | S1 |
| 38 | The program does not have any responses after clicking on "Save" button in the drop down menu, which should save the current grade and progress.                        | Correction: a pop-up message asking the user to save the file or not will be shown as code for "save as" is implemented. Swap code in "saveMenuItemActionPerformed()" function with that of original "saveAsMenuItemActionPerformed" function.   | Instructor Client GUI - Grade Test | S1 |
| 39 | The program does not have any responses after clicking on "Save As" button in the drop down menu, which should save the current grade and progress as a new file.       | A call for "save as" is, in fact, a "save" action. Swap code in "saveAsMenuItemActionPerformed" function with that of original "saveMenuItemActionPerformed()" function. (The user will not know whether the file is successfully saved or not though; display a message on whether the file has been successfully saved by adding relevant code after the line running "test.save(testFile);" will be useful) | Instructor Client GUI - Grade Test | S1 |
| 40 | The program does not have any responses after clicking on "Exit" button in the drop down menu, which should exit the program.   | The function "exitMenuItemActionPerformed()" has no any implementation. Implement the function by calling "System.exit(0)".  | Instructor Client GUI - Grade Test | S1 |

|    |  |   |                                    |    |
|----|--|---|------------------------------------|----|
| 41 | Clicking on the "Show summary" button will yield a miscalculated summary: the number of total points and percentage are incorrect. | For the numbr of total points, In line 67 of "Question.java" should be written as "this.ptsEarned += ptsEarned;" (refer 43). For the percentage, as its value is determined from line 47 of "summaryDialog.java", it is affected by the incorrect number of total points. Hence, correct the line 67 of "Question.java" will correct the percentage part as well (provided that "getPtsPossible()" function of "Question.java" is correct). | Instructor Client GUI - Grade Test | S1 |
| 42 | The default key answer is identical to the student answer.   | In "GraderTestify.java" "loadStudentAnswer()" function, "setText()" function should get the answer from "test" variable, not "key" variable. In other words, line 709 should be "answerTextArea.setText(test.getQuestion(current - 1).getAnswer());".   | Instructor Client GUI - Grade Test | S1 |
| 43 | The scores stored in the student exam file changes each time it is loaded into the program.  | In "Question.java" "setPtsEarned()" function, "ptsEarned" variable should be added, not reset. In other words, line 67 should be written as "this.ptsEarned += ptsEarned;".   | Instructor Client GUI - Grade Test | S1 |
| 44 | The default label next to "Language" is "jLabel3", which is not a valid language.  | Set the default value of setText() function to be one of the languages (JAVA, C, CPP), or a null string.  | Instructor Client GUI - Grade Test | S2 |
| 45 | Clicking on the "next" button changes to the previous question.  | In "InstructorTestify.java" line 482, "current++" instead of "current--".   | Instructor Client GUI - Grade Test | S2 |
| 46 | Clicking on the "previous" button changes to the next question.  | In "InstructorTestify.java" line 468, "current--" instead of "current++".   | Instructor Client GUI - Grade Test | S2 |
| 47 | The comments and choices text areas should be disabled(gray) when no test is loaded into the program.                              | Set text fields to disabled until a test is loaded  | Instructor Client GUI - Grade Test | S3 |

|    |  |  |                                    |    |
|----|--|--|------------------------------------|----|
| 48 | Clicking on the "down" arrow in the "Comments" text area will scroll the comments out of view.   | This is intentional, you can't see all comments at once in the small window                                      | Instructor Client GUI - Grade Test | S3 |
| 49 | The label for "student answer" should be more to the right and align with the text area below.   | In "GraderTestify.java" line 301, increase magnitude of the three arguments within ".addGap()".                  | Instructor Client GUI - Grade Test | S3 |
| 50 | The "Points Possible" should be a label instead of a text field to indicate that the value is unchangable.   | Change the variable into the label type rather than the text type.   | Instructor Client GUI - Grade Test | S3 |
| 51 | The "Language" label becomes gray when the question is of type -- programming. This does not make sense since the "C" label remains.   | The program should enable the language label if the question is of type programming.                             | Instructor Client GUI - Grade Test | S3 |
| 52 | There is no value next to the "compiles" label.  | The program does not have the function to check compilation. No value is displayed. It should add code to check. | Instructor Client GUI - Grade Test | S3 |
| 69 | Clicking on "Exit Test" button will bring up a file chooser. The button's message is counter intuitive. It should be something like "Start Test". To be more precise, it should be "Open Test" (from the last page). Furthermore, displaying a file chooser without noticing the users is not user-friendly. | On line 142, the openAction is called when it should be the close Action   | Student Client GUI                 | S3 |
| 53 | Clicking on "Grade Test" button will open the a screen to edit test.   | Swap action listeners for "Grade Test" and "Edit Test" buttons.  | Instructor Client GUI - Edit Test  | S1 |
| 54 | Next and previous buttons switched place.  | Swap setText() functions of the buttons.   | Instructor Client GUI - Edit Test  | S1 |
| 55 | When you edit the "Points Possible" box, it applied to all questions in the exam. That means every question will weight the same points.   | set the textfield to empty in function<br>newTestMenuItemActionPerformed   | Instructor Client GUI - Edit Test  | S1 |
| 56 | When you delete the "Choices" entry, it will reappear again even after you save it.  | In function<br>deleteChoiceButtonActionPerformed<br>locate saveQuestion() to the end of the function             | Instructor Client GUI - Edit Test  | S1 |
| 57 | When you add an entry in the "Choices" entry, there will be two identicals entries.  | delete the duplicated code<br>((MultipleChoice) question).<br>addChoice(newChoice);                              | Instructor Client GUI - Edit Test  | S1 |
| 58 | The number select options will not bring you to the correct question number. It will still remains at the current question after you clicked different question number.  | After the reconstructComboBox() is fixed in defect 61, this defect is fixed.                                     | Instructor Client GUI - Edit Test  | S1 |

|    |  |  |                                   |    |
|----|--|--|-----------------------------------|----|
| 59 | The "Answer" box will not update after you clicked next or previous button.  | In function saveQuestion(), add an if else statement to set the Answer of the question by using the setAnswer(answerTextArea.getText()). | Instructor Client GUI - Edit Test | S1 |
| 60 | The "Answer" box will not update even after you save it.   | In function saveQuestion(), add an if else statement to set the Answer of the question by using the setAnswer(answerTextArea.getText()). | Instructor Client GUI - Edit Test | S1 |
| 61 | After opening an existing exam file, the number select option showed nothing at the last entry. When you clicked it, it does nothing. However, after you click previous or next, it will showed the correct data but the last entry still showed blank in the number select options. | In the function reconstructComboBox(), the test.length()-1 in the for loop should change to test.length().                               | Instructor Client GUI - Edit Test | S1 |
| 62 | Refer to defect 61. After you add a new question, the number select option will display correctly.   | After the reconstructComboBox() is fixed in defect 61, this defect is fixed.   | Instructor Client GUI - Edit Test | S1 |
| 63 | After you delete an existing question, the number select option will not update. It will still display the deleted question number.  | In deleteChoiceButtonActionPerformed add code for reset the question numbers if deletion executed successfully.                          | Instructor Client GUI - Edit Test | S1 |
| 64 | The File and Language for the programming questions still showed their data in other question number but it has been greyed out.   | set the corresponding text to empty for functions next button listening function nextQButtonActionPerformed.                             | Instructor Client GUI - Edit Test | S3 |
| 65 | The Test -> Properties page needs to click "apply" twice to turn off the window.   | In showProperties(), the PropDialog class is being called twice. Remove the repetitive code.   | Instructor Client GUI - Edit Test | S3 |

|    |   |   |                                   |    |
|----|---|---|-----------------------------------|----|
| 66 | In the Test -> Properties, date box, when user changed the year, month and day data to an invalid data it will not pop up an error message. Instead it will automatically revert those data to a random correct formatted date. | Write an "if" statement to test whether the data inputted is invalid. If so, pop up an error message indicating an invalid data has been inputted, and set the year, month and day data to their default value. (In fact, automatically revert the invalid data to the random correct formatted date is acceptable. In this case, show a relevant message before reverting the invalid data is necessary to inform the user). | Instructor Client GUI - Edit Test | S2 |
| 67 | In the Test -> Properties, when I clicked the cancel button, it does nothing.   | The function "cancelButtonActionPerformed()" has no any implementation. Implement the function by calling dispose() to close the GUI window.  | Instructor Client GUI - Edit Test | S3 |
| 68 | In the Test -> Properties, I need to clicked the exit button on the top right twice in order to exit.   | In showProperties(), the PropDialog class is being called twice. Remove the repetitive code.  | Instructor Client GUI - Edit Test | S3 |
| 70 | There is no response after clicking on "Previous" and "Next" buttons on the student main page   | On lines 612 and 630 there are extra if statements that cause the next / previous button to skip a question on the exam for no reason.  | Student Client GUI                | S1 |
| 71 | "Exit test" button does not work at first click   | In function openActionPerformed delete the code "if(openBool == false) statement and its content  | Student Client GUI                | S1 |
| 72 | When the student choose a wrong file, the main page should prompt coresponding error instead of reminding user "please enter both last and first name "   | The only error message entered is the "please enter first..." error, no error message exists for any other error  | Student Client GUI                | S3 |
| 73 | The student click cancel button when he choose file, the system prompt a error "please enter both last and first name"  | The only error message entered is the "please enter first..." error, no error message exists for any other error  | Student Client GUI                | S3 |
| 74 | The last name can be empty when starting the test.  | Add a check for null last name like the one for a null first name   | Student Client GUI                | S1 |

|    |  |  |                    |    |
|----|--|--|--------------------|----|
| 75 | The name can have special characters, number or signs inside.  | Add a check for special characters after the null checks for first/last name   | Student Client GUI | S1 |
| 76 | The students can change the questions in the "Question Prompt" for short-answer type questions.  | Make the text field for question prompt uneditable   | Student Client GUI | S1 |
| 77 | Amount of points assigned for each question is incorrect. It is always 25.   | The point value is hard coded, change to reflect the point value specified when the test is created  | Student Client GUI | S1 |
| 78 | The first question is missed.  | The starting index is incorrect  | Student Client GUI | S1 |
| 79 | The students start at the second page. The third question is shown when the students just start the test. The students have to press "Previous" to access the second question.(Note that the first question is skipped.)                             | The starting index is incorrect  | Student Client GUI | S1 |
| 80 | when navigating to next questions, the answer inputted for a previous question will remain at the field.   | empty answer fields when the next button is clicked  | Student Client GUI | S1 |
| 81 | Several questions from behind are skipped.   | there are two if statements in student testify in the previous/next functions that should be removed. they skip the next question after any question with a number mod 3 | Student Client GUI | S1 |
| 82 | After the first submission, the students can still press "Previous" to go back to the questions.   | In "StudentTestify.java", remove line 469 to 474. The "Previous" button is unnecessary.  | Student Client GUI | S1 |
| 83 | Nothing happens when the students press "Open Test" in the last page. (The button should be "Exit Test" from the first page)   | correct the text shown on the button, add listener to close text   | Student Client GUI | S1 |
| 84 | Multiple submissions are allowed. The "Submit" button does not gray out after the first submission.  | This may be intended, otherwise test should exit after first submission, add exit listener to submit button  | Student Client GUI | S1 |
| 85 | If the students start the test with the programming type question, the question will change into a multiple choice question.   | This is due to a multiple choice answer set being hard coded as the first answer set. This needs to be changed to be dynamic   | Student Client GUI | S1 |
| 86 | When the students reach the last question of the test(if the program does not freeze), the "Next" button will change to "Submit". The students can press "Previous" to go back to the previous questions, but the button will remain to be "Submit". | Change the check for the submit button to see if it is equal to the end of the index, instead of current check   | Student Client GUI | S1 |

|    |   |  |                                |    |
|----|---|--|--------------------------------|----|
| 87 | Some questions field are empty and can be edited.   | Change question fields to be uneditable text fields  | Student Client GUI             | S1 |
| 88 | The page number is disordered. For instance, the students start at page 2, but navigating to next page will show a page number of 4.  | This is due to unnecessary if statement; remove the statement.   | Student Client GUI             | S2 |
| 89 | The compiler does not work for the programming question. It prompts an error message.   | On line 674 of studentTestify.java, Only compiles if there are an odd number of files. remove this if statement  | Student Client GUI             | S1 |
| 90 | If number of the questions exceeds a certain number, the students will not be able to submit their test when reaching the last question. The "Submit button" is not available. Instead, the "Next" button will remain. Pressing it yields nothing. Furthermore, the students cannot press "Previous". | This is due to unnecessary if statements in the next/previous method, they should be removed   | Student Client GUI             | S1 |
| 91 | The question and answer does not match each other. For example, a programming type question is shown with answer prompt of multiple-choice due to the disalignment.   | Correct starting array index so that questions and answers align   | Student Client GUI             | S1 |
| 92 | In the main page, there is no password field. This means that any student can log into another student's account.   | Add a password field, and check it against the password stored for the username  | Student Client GUI             | S1 |
| 93 | The students can start more than one test with the same first name and last name at the same time.  | Is this a bug? or should students be able to take multiple tests at once   | Student Client GUI             | S1 |
| 94 | The code of ExitMenuItem action performed did not exit the program  | added a statement to exit the program  | Instructor Edit/Grade Test GUI | S1 |
| 95 | Made the auto-grade option greyed out until a test is open  | added a setEnabled(false) initially and setEnabled(true) after test is opened  | Instructor Edit/Grade Test GUI | S1 |
| 96 | The jLabel for Language said jLabel3  | Removed the jLabel3 from the jLabel  | Instructor Edit/Grade Test GUI | S3 |
| 97 | jlabel2 said "Compile?"   | Got rid of the question mark and made it "Compile"   | Instructor Edit/Grade Test GUI | S3 |
| 98 | nextQButton and prevQButton would perform the wrong roles. nextQButton would make it go to a previous question while prevQButton would make it go to the next question  | switched the code of nextQButton for prevQButton and vice verse  | Instructor Edit/Grade Test GUI | S2 |
| 99 | The button "Open Student Exam" was enabled, but you needed to open the key first.   | Set openStudentMenuItem to be disabled until a key was opened. This fixed the crash if you try to open an exam before you open a key, and opening the exam twice | Instructor Edit/Grade Test GUI | S1 |



|     |  |   |                                |    |
|-----|--|---|--------------------------------|----|
| 100 | Comments text area was enabled even though there was nothing opened  | changed CommentsTextArea to be disable until a test is opened   | Instructor Edit/Grade Test GUI | S3 |
| 101 | Student Answer label was not aligned with the answer box   | Increased the gap size to make it be aligned with the text area for the answer  | Instructor Edit/Grade Test GUI | S3 |
| 102 | If you press the previous button from a multiple choice question, it would keep the answers even if the new question is a short answer | Made it so listModel is cleared every time you click previous or next   | Instructor Edit/Grade Test GUI | S2 |
| 103 | Summary would show incorrect calculations  | Changed it, so it no longer can give negatives points. As well as corrected the formula to calculate the proper grade   | Instructor Edit/Grade Test GUI | S1 |
| 104 | Auto-grade feature would crash, if there were non-multiple choices questions   | In the autogradeMenuItemActionPerformed made it skip to the next question if it was not a multiple choices question   | Instructor Edit/Grade Test GUI | S1 |
| 105 | Clicking "Save" and "Save As" would not acknowledge that it has been saved   | Added a Dialog window to pop up that a save has been successfull. Added it to open in the saveAsMenuItemActionPerformed and saveMenuItemActionPerformed functions | Instructor Edit/Grade Test GUI | S2 |