**Chapter 14 Book Work**

Logan Yeubanks

3. I don’t see any input validation so that could be a issue when getting details or adding them. It could also be a issue when sending the details to other places such as a server or another class.

4. Hash map can not be used because it would not be in a specific order to be used later.

5. Checked to see if user input was equal to a key and then printed information about that key.

6. If calculated directly there would be instances where any decimal would round to the closest whole number because they are integers.

7. It would probably be quite hard because new methods would need to be added to add and remove them and a new HashMap would be needed to store them.

8. a NullPointerException error occurs because the specific key can not be removed from contact details.

9. If some information is intended to be saved and a certain error can crash the program than all save data will be lost.

10. No it is not acceptable for a program that a customer purchased to crash upon a simple error. All known errors should be handled with appropriate exceptions before being sold to consumers.

11.

If(keyUsed(key)==true)

{

ContactDetails details = book.get(key);

Book.remove(details.getName());

Book.remove(details.getPhone());

numberOfEntries--;

12. It is necessary to report if a key is invalid. I would do so with a catch for a NullPointerException.

13. I would fix these errors the same way because they are necessary to fix. In this scenario I would keep using the NullPointerException because it is applicable.

14. It is not needed because the object returned is a ContactDetails object.

15. The errors that get thrown should be reported to the window and not just the terminal.

16. Try and catch blocks can be put throughout the project and print the type of error encountered.

17. Some ways that errors could be thrown are when different types such as integers are inputted in a Boolean or strings in an integer input.

18. The warning method is higher priority and is meant to be used in dangerous situations and shows how critical it is. The info method shows information to describe something and often just shows progress.

19. The GUI based version should have some sort of pop-up error warning while the text based one would just be reported in the terminal.

21. No it would not need an error notification because it would not throw a error if no matches were found.

22. Passing in the same name and address would be inappropriate and cause problems.

23. Constructors can also throw exceptions so the appropriate exception should be thrown and a report should be given that the new object could not be created.

24. FileNotFoundException, EOFException, IOException

25. It is a unchecked exception.

31. It will run but it is wrong because the print statement is written outside the catch block.

33. The super class exception is where the error would be caught so the code will terminate due to the error.

34.

try

{

Book.addDetails(new ContactDetails(name, phone, address));

}

catch (DuplicateKeyException e)

{

System.out.println(“Exception caught”);

System.out.println(e.toString());

}

35. It would be a checked exception because it is checking things before adding them to the contact list.

36. assert consistentSize () : “Inconsistent book size”

37. There is a assert error if adding information that already exists occurs.

38. After testing I do not believe that there should be an assert statement to check consistency of a contact list because there already are assert statements for this in addDetails and removeDetails.

39. It would run successfully because address is linked to name and phone number.

40. The consistency of ContactDetails would be ruined likely as the indexes would change and no longer match their related name and address.

41. There is information on file paths and how to add or read files.

42. isFile is used to point to a file and isDirectory is used to check for a directory.

43. No information about a file stored in a file object is able to be described.

49.

nextByte – converts next token of scanned input into a byte.

nextBoolean – converts next token of scanned input into type boolean and returns it.

nextLine – scans and returns the total current line until it finds the newline character.

53. There are no issues when a new field is added in.