CSCI 3901 LAB 4: GIT

Team Members:

Benny Tharigopala [B00899629]

Nadipineni Hemanth Kumar [B00899473]

PART - 1: History

1) using the “git status”, “git log”, and “git diff” commands, answer the following questions:

1. how many constants were added between versions 4 and 5?

Answer: **4**

```
private static final int BOLD = 3;
private static final int UNDERLINE = 4;
private static final int ITALICIZE = 5;
private static final int LISTITEM = 6;
```

2. In which version was the method “main” separated into its own file?

Answer: **Version 4**

```
“095bfc5 Split main() from the rest of the code.  Include code to split the body into paragraphs / lists”
```

3. how many versions of the code have been checked in? how can you tell?

Answer: **5 versions of the code have been checked in. git log origin revealed 5 different hash values.**

```
5f24d61 (HEAD -> master, origin/master, origin/HEAD) Include translation of pair-based formatting characters, notably bold, italisize, and underline
095bfc5 Split main() from the rest of the code.  Include code to split the body into paragraphs / lists
```

```
2b0934b Include formatting for the metadata section of the file.  Still print the body
as basic text.
7095699 Write enough code to run through the whole file and print it to the screen
aba7db8 Just ensure that I can get user input properly
```

4. In which version did the external documentation file appear in the repository?

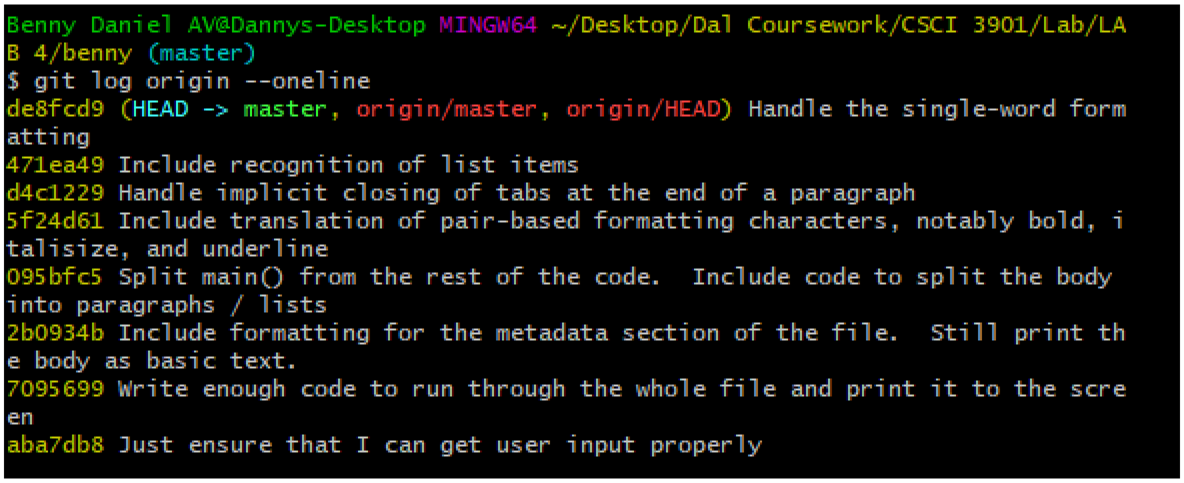
Answer: **Version 3** – commit 2b0934b7cc1c5c5331f8469ffd3f692ca4a6f941

PART - 2: Basic Operations

Add the changes from the directory v6 to the repository and commit the changes to the central repository.

Include an appropriate message for the update. Do the same again with the files from the v7 and v8 directories and commit each of the changes to the central repository so that they appear as distinct checkpoints in the repository.

Validation:



```
Benny Daniel AV@Dannys-Desktop MINGW64 ~/Desktop/Dal Coursework/CSCI 3901/Lab/LA
B 4/benny (master)
$ git log origin --oneline
de8fcd9 (HEAD -> master, origin/master, origin/HEAD) Handle the single-word form
atting
471ea49 Include recognition of list items
d4c1229 Handle implicit closing of tabs at the end of a paragraph
5f24d61 Include translation of pair-based formatting characters, notably bold, i
talisize, and underline
095bfc5 Split main() from the rest of the code.  Include code to split the body
into paragraphs / lists
2b0934b Include formatting for the metadata section of the file.  Still print th
e body as basic text.
7095699 Write enough code to run through the whole file and print it to the scre
en
aba7db8 Just ensure that I can get user input properly
```

PART - 3: Branch

Create a new branch called “new_feature”.

Your new branch should be off your repository’s master branch.

Add the changes from directory v8.1 to this new branch and commit the changes.

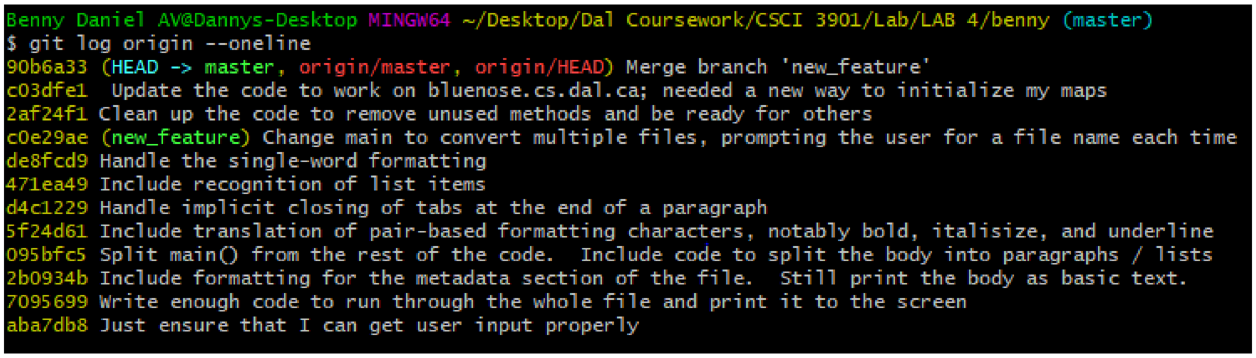
Add the changes from the directory v9 to your master branch, not the “new feature” branch. Commit the changes.

Add the changes from the directory v10 to your master branch and commit the changes.

Finally, merge the changes from your “new_feature” branch into your original branch and commit the changes.

The output to assess your outcome of these operations will be the git structures themselves that appear in the central repository.

Validation:

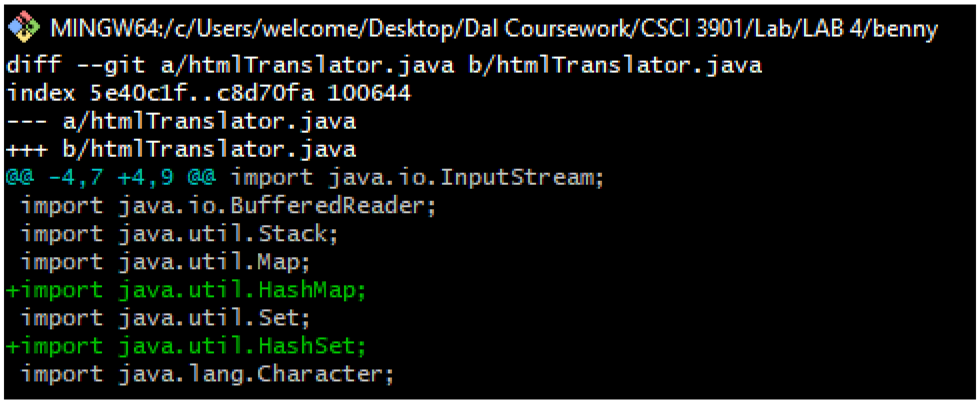


```
Benny Daniel AV@Dannys-Desktop MINGW64 ~/Desktop/Dal Coursework/CSCI 3901/Lab/LAB 4/benny (master)
$ git log origin --oneline
90b6a33 (HEAD -> master, origin/master, origin/HEAD) Merge branch 'new_feature'
c03dfe1  Update the code to work on bluenose.cs.dal.ca; needed a new way to initialize my maps
2af24f1 Clean up the code to remove unused methods and be ready for others
c0e29ae (new_feature) Change main to convert multiple files, prompting the user for a file name each time
de8fcd9 Handle the single-word formatting
471ea49 Include recognition of list items
d4c1229 Handle implicit closing of tabs at the end of a paragraph
5f24d61 Include translation of pair-based formatting characters, notably bold, italisize, and underline
095bfc5 Split main() from the rest of the code.  Include code to split the body into paragraphs / lists
2b0934b Include formatting for the metadata section of the file.  Still print the body as basic text.
7095699 Write enough code to run through the whole file and print it to the screen
aba7db8 Just ensure that I can get user input properly
```

Use Git to answer the following questions:

5. Which imports were added between versions 9 and 10?

Answer: **The HashMap and HashSet imports were added between versions 9 and 10.**



```
MINGW64:/c/Users/welcome/Desktop/Dal Coursework/CSCI 3901/Lab/LAB 4/benny
diff --git a/htmlTranslator.java b/htmlTranslator.java
index 5e40c1f..c8d70fa 100644
--- a/htmlTranslator.java
+++ b/htmlTranslator.java
@@ -4,7 +4,9 @@ import java.io.InputStream;
 import java.io.BufferedReader;
 import java.util.Stack;
 import java.util.Map;
+import java.util.HashMap;
 import java.util.Set;
+import java.util.HashSet;
 import java.lang.Character;
```

Broadening Questions:

When working alone on a project, how frequently should you commit your code to a version control system? Explain why.

Answer: I should commit my code whenever I get to a state that I would want to get back to, or whenever I reach a checkpoint relevant to the feature I am working on.

Why?

1. If I inadvertently break my code while creating new features, I can roll back to my last checkpoint with ease.
2. When I build several features and make multiple changes, commits are useful in determining what changed since the last known checkpoint.
3. Frequent commits can help me to maintain a history of what was developed.

When working in a team, how frequently should you commit your code to a version control system? Explain why.

Answer: While working in a team, I should commit my codes as regularly as I can. In the case of a team, the room for error and conflicts is higher. Therefore, committing codes frequently can reduce the chances for conflicts.

In addition, frequent commits ensure that others understand what's being developed, with a single look at my log.

Why might you create branches for your project in your version control system?

Answer: I might create branches for one or more of the following reasons:

1. To isolate a new feature that I am developing from an existing code which functions well.
2. When I would like to test out multiple approaches for a feature and would like to try them in parallel instead of creating multiple files.
3. To include and remove features without affecting the master branch.

REFERENCES:

1. https://www.youtube.com/watch?v=mrkf3w_2fbo&feature=youtu.be
2. https://www.youtube.com/watch?v=iaszpmz2thw&feature=youtu.be
3. https://youtu.be/buhcgpbtyno